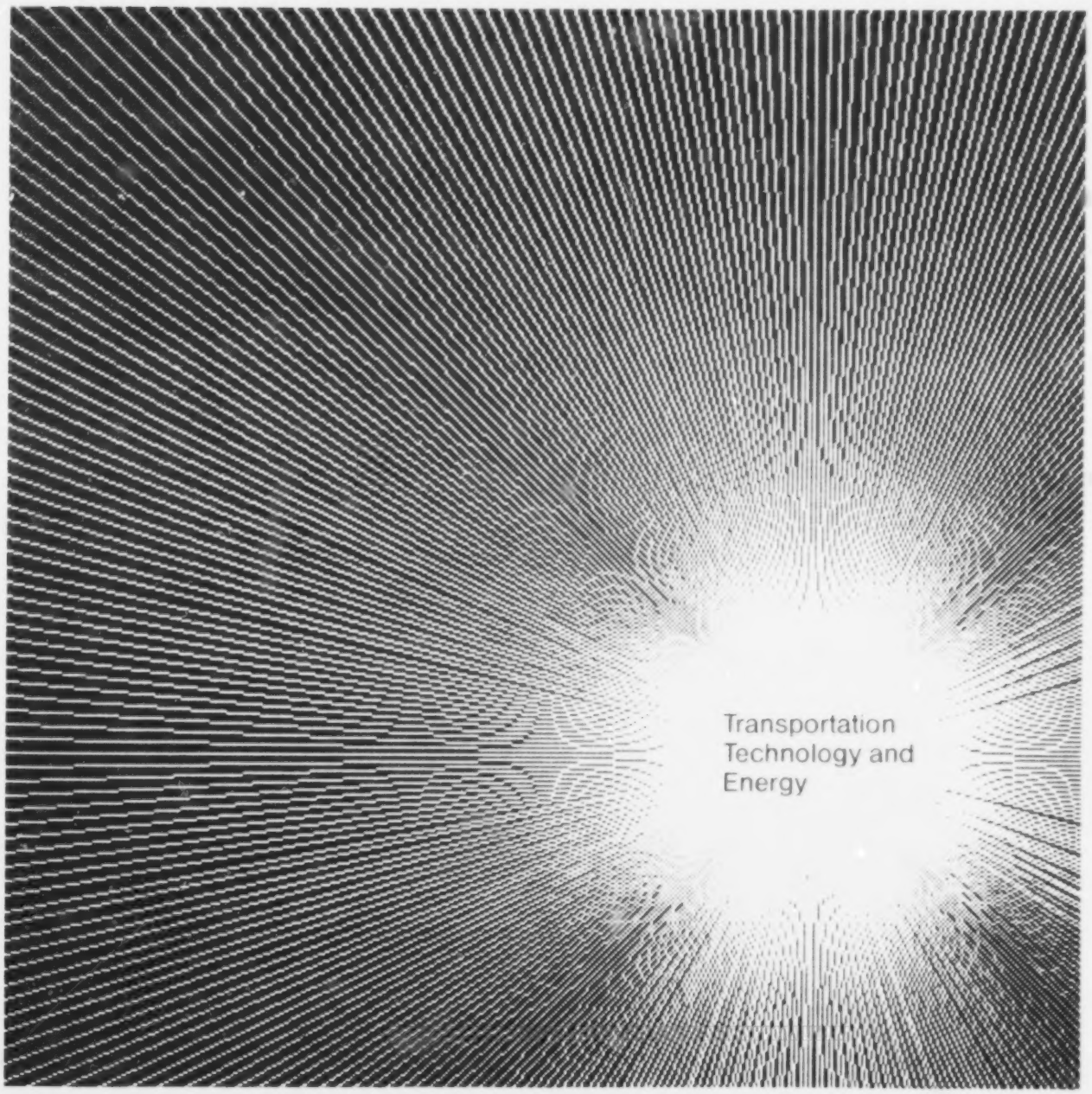
Ontario

Ministry of
Transportation and
Communications

Transportation
Technology and
Energy Branch

Conceptualization of a Mobile Robot for Cleaning of Transit Vehicles

TCT-86-01



TCT-86-01

Conceptualization of a Mobile Robot for Cleaning of Transit Vehicles

Prepared by:
C. Jaluba
Consultant

Prepared for:
W. Wierclenski
Project Manager

Published by:
Transit Control Technology and Systems Office
Transportation Technology and Energy Branch
Ontario Ministry of Transportation and Communications
Hon. Ed Fulton, Minister
D.G. Hobbs, Deputy Minister

Published without prejudice
as to the application of the findings.
Crown copyright reserved; however, this
document may be reproduced for non-commercial
purposes with attribution to the Ministry.

For additional copies, write:
The Editor, Technical Publications
Ontario Ministry of Transportation and Communications
1201 Wilson Avenue
Downsview, Ontario
Canada M3M 1J8
(416) 248-7226

December 1986

Technical Report Documentation Page

MTC Report Number TCT-86-01	ISBN Number:	Other Document Number:	Date: Dec. 1986

Title: Conceptualization of a Mobile Robot for Cleaning of Transit Vehicles

Authors (name, title and/or organization):
C. Jaluba, Consultant

Published by:
Transit Control Technology and Systems Office
Transportation Technology and Energy Branch, MTC

Client Group:

Participating Agencies:

Abstract:
The conceptualization and design of an autonomous robotic cart equipped with two controllable cleaning tools is described in this report.

The following areas are addressed:
- selecting dc gearmotors, including the dynamic equations of the robot;
- cart control system and theory of operation of the robot;
- robotic information flow and software architecture of the robot;
- motion control, including trajectory planning and kinematic equations of the mobile cart and the cleaning tools, with consideration given to the task of cleaning rounded corners.

More work is planned on the trajectory error correction and guidance system software at a later date.

Key Words: mobile robot, equations of motion, motion control strategy, trajectory planning

Comments:

Distribution: by request

Copyright Status: Crown copyright

DISCLAIMER

This report was written by C. Jaluba, a consultant, and is published as a public service. The Ministry of Transportation and Communications does not, however, warrant the accuracy of its contents and cannot guarantee or assume liability for the effectiveness of the devices and processes described in the report or that their use does not infringe on privately owned rights.

TABLE OF CONTENTS

	PAGE
1/ INTRODUCTION	1
2/ TECHNICAL OVERVIEW	2
2.1/ Major Subsystems of the Autonomous Mobile Robotic Cart.	2
2.1.1/ Overall Configuration of the Cart Control System	2
2.1.2/ Theory of Operation of the Mobile Cart	2
2.1.3/ Robotic Information Flow	3
2.2/ Software Architecture of the Robot.	4
3/ SELECTING DC GEARMOTOR/ENCODER FOR THE DRIVES AND VACUUM ARM	7
3.1/ Overall Configuration of the Robotic Control System	7
3.2/ Assumptions	8
3.3/ Description of the Cart Geometry	8
3.4/ Mathematical Model of the Mobile Platform	9
3.5/ The Impact of the Cart Kinematic Requirements on the Drive.	16
3.6/ Parameters Necessary for the Calculation of the Drive Motors.	17
3.6.1/ Cart Moment of Inertia About Its Centre of Mass.	17
3.6.2/ Motor Armature Polar Moment of Inertia and Viscous Coefficient of Friction.	17
3.6.3/ Angle of Inclination α	18
3.6.4/ Geometrical Parameters	18
3.6.5/ Kinematic Parameters	18
3.7/ Calculation of the Motor Torques.	19
3.8/ Motor Choice.	20
3.9/ Encoder Choice.	21
4/ MOTION CONTROL STRATEGY	23
4.1/ Cleaning Robot Configuration.	23
4.2/ Robot Functions	23
4.3/ Theory of Operation	26
4.3.1/ Hardware Organization.	26
4.3.2/ Robot Operation.	27
4.4/ Software Architecture	29
5/ MOTION CONTROL -- TRAJECTORY PLANNING MODULE	32
5.1/ Introduction	32
5.2/ Trajectory Steps and Pseudocode	32
5.2.1/ Trajectory Steps	32
5.2.2/ Pseudocode of the Trajectory	33

TABLE OF CONTENTS (CONT'D)

PAGE

5.2.3/ Kinematic Equations of the Robotic Cart. 34
5.3/ Task Steps and Pseudocode 35
5.3.1/ Floor Coverage 35
5.3.2/ Kinematic Equations of the Main Tool 36
5.3.3/ Pseudocode of the Task Steps 41
5.4/ Integration 47
5.5/ Motion Control Strategy Using DMC-200 Controller 47

6/ CONCLUSIONS 48

7/ RECOMMENDATIONS 49

8/ REFERENCES 50

APPENDIX 71

LIST OF FIGURES

	PAGE
1/ Block Diagram of the Motion Control System	55
2/ Logical Layout of the Robotic Circuits	56
3/ Robotic Information Flow	57
4/ Software Architecture of the Robot	58
5/ First Logical Level of MRB	59
6/ Geometrical Description of the Robotic Cart.	60
7/ Straight-Line Motion	61
8/ Rotation in Place (About Pivoting Point P)	62
9/ General Motion Case.	63
10/ DC Motor Loaded with the Inertial Load, m_L	64
11/ Subway Car Schematic.	65
12/ Working Envelope, Cart Trajectory	66
13/ Motion Characteristics of the Robotic Cart.	67
14/ Working Envelope, Floor Covering.	68
15/ Corner -- Concave Circle.	69
16/ Corner -- Convex Circle	69
17/ Discretization of Motion for the Concave Corner, Segment by Segment.	70

EXECUTIVE SUMMARY

A recent study conducted by MTC showed that maintenance and inspection of transportation systems is expensive. The task of cleaning transportation vehicles is unpleasant and inefficient. The use of cleaning robots seems to be a reasonable solution, with robotic mobility a desirable quality. Robot flexibility may be achieved with little effort by supporting a manipulator on a small robotic cart. Controllable cleaning tools may be attached to the mobile cart for the task of floor cleaning. The prototype system would be guided intelligently rather than relying on a wire-guided system or operator.

As the magnitude and complexity of such a project is considerable, it was decided to start with the conceptualization and design of an autonomous robotic cart equipped with two controllable cleaning tools.

The mechanical design of the robotic cart is developed for a four-wheeled, square robotic cart. Two wheels mounted at the front of the cart are driven by dc motors, and the two remaining wheels are casters, such that the cart can travel in any arbitrary direction. The two cleaning tools consist of two sliding arms. The power supply unit is on board and consists of a propane motor and rechargeable batteries. The considerable amount of equipment condensed on a relatively small surface makes the design of the cart difficult.

The associated control system consists of two DMC-200 two-axis controllers, which communicate with an on-board host computer. The motion performed by the cart is a plane motion. The motion control commands for the mobile cart and the cleaning tools are issued by the host computer, which has the trajectory stored in its memory.

The mathematical model of the robot consists of task planning, kinematics, and control of the robotic cart and cleaning tools. This model is needed for the development of software required for the robot operation.

The guidance system consists of ultrasonic distance sensors. The distance sensors provide the actual position of the cart. The preprogrammed position of the cart is compared to the actual position. If the trajectory deviation exceeds a required tolerance, then a corrective action is taken. The corrective action restores the cart's planned trajectory.

This report addresses the following areas:

- selecting dc gearmotors, including the dynamic equations of the robot;
- cart control system and theory of operation of the robot;
- robotic information flow and software architecture of the robot;
- motion control, including trajectory planning and kinematic equations of the mobile cart and the cleaning tools, with consideration given to the task of cleaning rounded corners.

More work was planned on the trajectory error correction and guidance system software, but it was left for a later stage.

The conclusions of this work are given in Chapter 6.

The tasks which need to be completed for the development of the robot artificial intelligence are given in Chapter 7.

Currently, the mobile cleaning robot is in the engineering prototype phase.

1/ INTRODUCTION

The mathematical model of the autonomous mobile robotic cart is developed for a four-wheeled robotic cart. Two wheels are mounted on the front of the cart and are driven by dc motors. The two remaining wheels are castors such that the instantaneous motion has only two degrees of freedom. Since each front wheel is driven independently, three types of cart trajectories may be obtained:

1/ a straight line
2/ a circular trajectory
3/ a rotation about the midpoint between the front wheels

These trajectory types are more than sufficient for cart movement in any arbitrary trajectory.

The cart guides itself on a preprogrammed path. Distance sensors provide the actual position of the cart. The preprogrammed position of the cart is compared to the actual position. If the trajectory deviation exceeds a required tolerance, then a corrective action is taken to restore the cart planned trajectory. At the same time, the robotics processing unit (on-board host computer) monitors the planned task(s) and other autonomous functions, such as interruptions from the emergency micro-switches, brakes, and power supply units.

2/ TECHNICAL OVERVIEW

2.1/ Major Subsystems of the Autonomous Mobile Robotic Cart

2.1.1/ Overall Configuration of the Cart Control System

The control strategy is suitable for a mobile robot equipped with two moving controllable cleaning tools. For the cart motion, a two-axis control system is required to co-ordinate x-y plane motion (DMC-200). The controller communicates with an on-board host computer, which issues commands "on the fly" [1]. An encoder is attached to the shaft of each of the two driving motors to ensure the feedback of the two closed control loops. The gearbox backlash may be eliminated by the software, using a homing command for the driving wheels under the brakes. An alternative solution is based on the harmonic-drive gearbox with zero backlash. With a DMC-200 controller, the velocity feedback and the cross-coupling between the driving wheels may be achieved from the controller software. With such a configuration, tachometers and cross-coupling circuits are redundant. A control scheme for the motion of the mobile cart is shown in Figure 1.

2.1.2/ Theory of Operation of the Mobile Cart

The mobile robotic cart is provided with three power supply units:
1/ a propane motor for the vacuum pump linked to the cleaning tools;
2/ a main battery unit for electrical motors, control circuits, guidance system, and host computer;
3/ a secondary battery unit for an emergency circuit that is separate from the main battery power unit.

The logical layout of the robot circuits is shown in Figure 2. The START/STOP PUSH BUTTON turns on/off both main and emergency circuits. Thus, if this button is off, the emergency may not be signalled. The solid-state relay (SSR) is normally open. Two conditions may cause the interlock circuit to close the SSR: a power failure in one of the main circuit subsystems or a catastrophic trajectory error which is signalled by the main computer. For each interlock condition, an appropriate indicator lamp is lit. After the emergency condition is removed, pressing the RESET BUTTON resets the interlock. The robot command is taken over by the host computer software, which displays instructions on the screen of the monitor. If the source of the emergency is the computer power failure, then the data from the main computer memory are lost.

To make the robot fully operative, two modes of operation are provided from the robotic software: local and autonomous. The local mode is

useful for robot homing operation and for a catastrophic situation, while the autonomous mode is provided for the rest of the robot life. Both modes are implemented in the software through a series of menus displayed on the screen of the monitor. The description of the most important menus is given in Section 2.2.

2.1.3/ Robotic Information Flow

The robotic information flow addresses the robot functions. Figure 3 shows the logical flow of information required by the robot software. A brief description of each module, shown in Figure 3, is given as follows:

- Trajectory planning converts user's input (robot path description and tool's required motion) into a table (Table 1) which consists of trajectory segments necessary to accomplish the motion. The cart motion is characterized by location (position and orientation), speed (linear and angular velocity), and acceleration.

  The trajectory checkpoints are defined for the guidance. At each trajectory checkpoint, the host computer takes information from the guidance system and compares the actual cart position to the planned one.

- Inverse kinematics converts Table 1 into Table 2, which consists of commands to the motors. The commands may be expressed in encoder quadrature counts for the motors. These commands are later implemented through the programmable controller DMC-200 "on the fly."

  For each structured environment and required tasks, a new Table 2 may be created off-line and then transferred to the robot host computer. The trajectory planning and inverse kinematics modules may be implemented in any computer high-level language.

- The guidance system consists of distance ultrasonic sensors, and their readings are stored in some local buffers. Since, in some situations sensors may interfere with each other, giving false readings, the host computer turns on/off sensors selectively. The active sensors are triggered 30 to 60 times/min, such that the new reading updates the old one using some temporary buffers. When a trajectory checkpoint is encountered, a read statement is issued to take the data from the guidance buffers.

- The trajectory error correction module establishes a strategy of the trajectory error handling, which issues appropriate commands to the controllers. The idea is to bring the robot to a trajectory checkpoint from which the robot may continue its useful task.

With more sophisticated sensors, a better trajectory error correction module may be developed and better robot autonomy achieved.

- The mobile robot brain (MRB) monitors and controls processes and interruptions from the robotic system (Figure 3). MRB faces three situations:
  1/ interacts with the operator in local mode
  2/ interacts with Table 2 in autonomous mode
  3/ stays dormant during an emergency situation

A more detailed description of the MRB unit is given in Section 2.2.

2.2/ Software Architecture of the Robot

The software architecture depends on the robot application. However, a good part of this robot software applies to most of the robots with mobility.

The organization of the robot operating system is shown in Figure 4. Trajectory planning and inverse kinematics modules are implemented off-line, and their output file (Table 2) contains the environment and task description. The file comes to the host computer on a floppy disk as a preprocessor. Such a software organization permits the creation and storage of a versatile robotic data base, leaving the robotic operating system unchanged.

Routine jobs, high-priority real-time jobs, or special jobs should all be performed at the same time. A technique for handling numerous routines or programs simultaneously by overlapping or interleaving their execution is "multiprogramming." Priority multiprogramming is oriented toward concurrent operation of several types of applications. For example, the first priority level can be reserved for a program that must provide rapid responses to real-time devices. Inside this real-time program, a scheduling plan or formula is established. Such a schedule is based on the list of priorities mentioned in Section 4.4. Interruptions may be handled using "an interrupt vectoring": polling sequence services interrupt requests, such that the value of each interruption request can be used as a system address to transfer control for the appropriate response routine.

The main program, MRB, initializes program parameters, resets computer memory, and controls the execution of three robot operating modes:
1/ autonomous
2/ local
3/ dormant

From the programming technique point of view, MRB consists of subordinate executable modules that are programmed using a set of macro-instructions recognized by the computer operating system. Each subordinate executable file has a structure similar to that of the MRB program. The third programming level is written in a high-level computer language or Assembler.

The program RUN controls the robot in the autonomous mode. It consists of a scheduler, converter (converts the input file, Table 2, into DMC-200 commands), interrupt handler, and a series of modules which produces the autonomous robot functions defined in the robotic information flowchart (Figure 3).

The program LOCAL controls the robot in the local mode. In the local mode, the host computer communicates with an operator, rather than using data from the file which contains Table 2. The program LOCAL consists of a scheduler, converter (converts the operator's input into DMC-200 commands), and an interrupt handler.

In the dormant mode, the robot is at rest and the program checks the status of the interlock circuits to see if the emergency situation has been removed. If it has, the main menu is displayed on the monitor.

Figure 5 shows the structure of the highest level of the software architecture (program MRB). This program communicates to the operators through a series of menus. When the operator turns the START/STOP PUSH BUTTON (Figure 2) on, the computer responds by displaying the main menu:

RME - reset memory
RUN - run the robot in autonomous mode
LOCAL - run the robot in local mode

ENTER:

The RME option resets the computer memory. This command is useful when the operator wants to restart the robot after an emergency situation has been removed.

The RUN option assigns the autonomous mode to the robot.

The LOCAL option assigns the local mode to the robot. In this mode the control of the driving motors and tools is under the operator's command. The operator is questioned by the computer program through a series of menus, and as an answer, the operator enters options and commands. The commands instruct the robot motors to move. The local mode is useful to

bring the robot and its tools to a homing, or any other desired, position.

Starting from the third programming level (Figure 4), the robot software should be built from the base of the software architecture.

3/ SELECTING DC GEARMOTOR/ENCODER FOR THE DRIVES AND VACUUM ARM

3.1/ Overall Configuration of the Robotic Control System

Two dc motors are provided to drive two independent driving wheels of a four-wheeled square cart to accomplish a predetermined trajectory. The two remaining wheels are casters. A two-axis control system is required to co-ordinate x-y plane motion (DMC-200) of the robotic cart. The controller communicates with a host on-board computer, which issues commands. The commands may be down-loaded from the host computer to the controller "on the fly" by selecting the appropriate MULTIBUS mode (shunting the MULT jumper on the DMC-200). Commands are transferred to the controller one character at a time using ASCII code.

The encoders provide the feedback to the two closed control loops. The encoders should be mounted on the motors shafts rather than on the driving wheels to achieve a stable control system. No combination of motor-encoder-drive wheel can avoid gearbox backlash, because from the measurements of driving wheel motion, there is no way to ascertain the position of the motion shaft in the backlash zone.

A velocity feedback always introduces a derivative factor to the control. It does not change the steady-state error, rather it improves the stability (anticipates the error change), but at the same time it amplifies noise. With a DMC-200 two-axis multibus dc motor controller, the velocity feedback is simulated using the encoder measurements, such that a velocity sensor is redundant. In addition, any stability of the control system may be achieved using the compensation parameters of the filter (DMC-200 instructions: GM-gain, ZR-zero, and PL-pole). Practically, the filter produces a phase lead by the zero, ZL, and the pole, PL. Not only the velocity feedback is simulated by DMC-200, but any cross-coupling between the two driving motors may be managed through a program implemented on the host computer. Thus, with such a configuration, tachometers and cross-coupling electronic circuits are redundant. A control scheme suitable for the robotic motion control, which fits with the DMC-200 controller, is shown in Figure 1. Micro-switches stop motors and activate brakes.

Since the cart configuration cannot be completely accurate at this design level, some assumptions are made to have enough data for the calculations of the requirements of the driving motors.

3.2/ Assumptions

The total cart load, its distribution, and its rolling coefficient of friction on wheels may be exactly determined at a latter stage, when all the equipment of the robot is ready to be mounted. Also, all frictions may be measured on the engineering prototype. However, since the drive motors should be decided at this project stage, some assumptions are made. The assumptions reflect the worst possible case of the robotic cart operation so that the drive motors will be suitable for any change in the design solution of the cart. The following assumptions are made:

- The maximum total weight of the cart with its equipment is 200 kg.
- The centre of mass of the cart is between the axle of the drive wheels and the axle of the casters, in the cart mid-plane. This assumption is justified because the traction of the motors is better when the centre of mass of the total load is closer to the axle of the driving motors.
- The mass of the cart is uniformly distributed, which is worse than the real cart configuration case, which has the heavy power supplies near the driving wheels.
- The coefficient of the rolling friction on the wheels is 0.01; it can change by the wheel radius and the type of material used for tires, as well as by the condition of the car floor.
- No lateral slip of the driving wheels is allowed; if the classic casters do not behave well, causing disturbances of the cart motion, a new caster design is suggested: two rotational axles of each castor and lateral springs on both sides of the wheeling casters.
- The maximum cart speed is 0.5 m/s, and the driving wheel radius is R_B = 0.0825 m.
- The slew speed of 0.5 m/s is reached in 0.5 s; the same requirement applies to the deceleration time.
- The cart suspension is rigid and the cart motion is a plane motion.
- The maximum inclination of the plane of motion with respect to the horizontal plane is $\sin \alpha \simeq 0.01$.
- In the dynamic equations of motion, the offset, e, of the free-wheeling casters is ignored since it is much smaller than the distance between drive wheels and casters, namely "$d_1 + d_2$."

3.3/ Description of the Cart Geometry

The engineering prototype of the mobile robotic cart consists of a four-wheeled robotic cart equipped with two tools for cleaning. Two wheels are mounted at the front of the cart and are driven by dc motors. The two remaining wheels are casters such that the instantaneous motion has only two degrees of freedom. Since each front wheel is driven independently, three types of cart trajectories may be obtained:

1/ a straight line
2/ a circular trajectory
3/ a rotation about the midpoint between the front wheels

These trajectory types are more than sufficient for cart movement in any arbitrary trajectory.

The approximate metric dimensions of the square cart are shown in Figure 6.

The assumption that the centre of mass of the load lies in the cart mid-plane is reasonable, because it is unlikely that an eventual lateral small offset of the centre of mass will be significant for the motor torques. Other possible side effects of the eventual offset of the centre of mass will be compensated by the feedback of the controller used for the cart motion control.

3.4/ Mathematical Model of the Mobile Platform

The two driving motors must be able to
- overcome any frictional load in the system; and
- start and stop all inertial loads, including that of its own motor.

To find the inertial load applied to each driving motor, it is necessary to establish dynamic equations of the cart motion. The set of the differential equations represents the mathematical model of the cart. In the equations of motion, inertia and friction terms appear clearly, and the information needed for the choice of the motors is defined. The set of dynamic equations is obtained for three different motion cases:

1/ Straight-Line Motion

Using Figure 7, the d'Alembert principle gives:

$$D_1(t) = D_2(t) = \frac{1}{2}\,(m_p a(t) + \mu m_p g) \qquad (1)$$

2/ Turning in Place (cart rotation about the midpoint P)

Using Figure 8, d'Alembert principle gives:

$$\pm D_1(t) = \mp D_2(t) = \frac{1}{2}\left[\frac{1}{d_3}\,(\circledS I_{zz} + m_p d_2^2)\varepsilon(t) + \mu m_p g\left(\frac{d_1}{d_1 + d_2} + \frac{d_2}{d_3}\right)\right] \qquad (2)$$

3/ Following a Curve of Instantaneous Radius R(t)

Using Figure 9, the d'Alembert principle gives:

$$D_1(t) = \frac{R(t) - d_3}{2(R^2(t) + d_3^2)} \left\{ \left[\circledcirc I_{zz} + m_p(R^2(t) + d_2^2) \right] \varepsilon(t) + \mu m_p g \frac{d_1 P + d_2 \sqrt{R^2(t) + (d_1 + d_2)^2}}{d_1 + d_2} \right\} \quad (3)$$

$$D_2(t) = \frac{R(t) + d_3}{2(R^2(t) + d_3^2)} \left\{ \left[\circledcirc I_{zz} + m_p(R^2(t) + d_2^2) \right] \varepsilon(t) + \mu m_p g \frac{d_1 P + d_2 \sqrt{R^2(t) + (d_1 + d_2)^2}}{d_1 + d_2} \right\}$$

An interesting case of these equations is for $R(t) = \pm d_3$ when the cart rotates about one of the driving wheels. In such a case, one wheel is at the rest while the other one is the most loaded. For $R(t) = d_3$, from Equation 3, it follows:

$$D_1(t) = 0$$

$$D_2(t) = \frac{1}{2d_3} \left\{ \left[\circledcirc I_{zz} + m_p(d_3^2 + d_2^2) \right] \varepsilon(t) + \mu m_p g \frac{d_1 d_3 + d_2 \sqrt{d_3^2 + (d_1 + d_2)^2}}{d_1 + d_2} \right\} \quad (4)$$

In the Equations 1 to 4, the first term is the inertia term, and the second one is the friction term.

The symbols used in the dynamic equations of motion are defined as follows:

P — point located at the intersection of the driving axle and the cart centreline

d_1 — distance from the rear free-wheeling casters to the cart centre of mass, measured in the mid-plane

d_2 perpendicular distance from the front axle to the centre of mass

d_3 distance between platform centreline and driving wheels

p coefficient defined as follows:
= d_3 if $R(t) < d_3$ (the pivoting point P is between the driving wheels)
= $R(t)$ if $R(t) > d_3$ (the pivoting point P is outside driving wheels)

$R(t)$ instantaneous radius of curvature of trajectory at point P

m_p total mass of the cart

I_{zz} platform moment of inertia about z-axis with respect to the centre of mass

μ rolling coefficient of friction on wheels

g gravitational acceleration, 9.81 m/s^2

$a(t)$ cart linear acceleration

$\varepsilon(t)$ cart angular acceleration

$D_1(t)$ applied force by the left motor to the left driving wheel

$D_2(t)$ applied force by the right motor to the right driving wheel

I_{zz} is defined using an uniform load distribution of the cart:

$$I_{zz} = \int (x^2 + y^2)dm \qquad (5)$$

If the load distribution is discrete, then:

$$I_{zz} = \sum_{i=1}^{n} (x_i^2 + y_i^2)m_i$$

where $(x_i y_i)$ are the co-ordinates of masses m_i, $i = 1,n$ and $\sum_{i=i}^{n} m_i = m_p$

The acceleration of the motor shaft after the gearbox is given in the following equations:

1/ Straight-Line Motion

$$\ddot{\theta}(t) = a(t)/R_B$$

$$\dot{\theta}(t) = v(t)/R_B \qquad (6)$$

2/ Turning in Place

$$\ddot{\theta}(t) = \varepsilon(t)d_3/R_B$$

$$\dot{\theta}(t) = \omega(t)d_3/R_B \qquad (7)$$

3/ General Case

$$\ddot{\theta}(t) = \varepsilon(t)(R(t) + d_3)/R_B$$

$$\dot{\theta}(t) = \omega(t)(R(t) + d_3)/R_B \qquad (8)$$

4/ Rotation About One of the Driving Wheels

$$\ddot{\theta}(t) = \varepsilon(t)2d_3/R_B$$

$$\dot{\theta}(t) = \omega(t)2d_3/R_B \qquad (9)$$

where $v(t)$ and $\omega(t)$ are the linear and angular velocities of the cart.

The load torque after the motor gear reduction is calculated for each drive motor, as follows:

$$m_{L1} = R_B D_1 = R_B(D_1 \text{ inertial} + D_1 \text{ frictional}) \qquad (10)$$

$$m_{L2} = R_B D_2 = R_B(D_2 \text{ inertial} + D_2 \text{ frictional})$$

As the drive motors are identical, it is enough to calculate only one of them.

The dynamic equation of motion of the rotor of the motor is obtained using Figure 10, as follows:

$$J_m \ddot{\theta}_m(t) + C_m \dot{\theta}^1_m(t) + m^1_L(t) = m^1_t(t) \qquad (11)$$

where

$\theta(t)$ = position of the drive motor shaft before reduction

J_m = armature polar moment of inertia

C_m = viscous coefficient of friction

$m_L^1(t)$ = load torque before the reduction

$m_t^1(t)$ = total motor torque between the motor and gearbox

Because the type of the dc motor to be used comes with the gearbox, Equation 11 is rewritten after the reduction, as follows:

$$m_t(t) = \frac{J_m}{k}\ddot{\theta}(t) + \frac{C_m}{k}\dot{\theta}(t) + m_L \qquad (12)$$

where

$\theta(t)$ = position of the wheel shaft
m_L = load torque as defined in Equation 10
m_t = total motor torque defined after the motor gear reduction
k = gear ratio of the drive motor, $k > 1$

As shown in Equation 10, $m_t(t)$ from Equation 12 may be defined as the sum of the inertia term, friction term, and a new, introduced, gravitational term due to inclination of the plane of motion with respect to the horizontal plane by angle of inclination α:

$$m_t(t) = \frac{J_m}{k}\ddot{\theta}(t) + \frac{C_m}{k}\dot{\theta}(t) + m_{L\ inert} + m_{L\ fric} + \frac{1}{2} m_p g \sin\alpha\ R_B, \text{ uphill movement} \qquad (13)$$

or

$$m_t(t) = m_{t\ inert} + m_{t\ fric} + m_{t\ grav} = m_{t\ inert} + m_r \qquad (14)$$

where m_r is the rated torque and m_t, the peak torque.

1/ Straight-Line Motion

From Equations 1, 6, 10, 13, and 14:

Peak torque:

$$m_t(t) = \frac{1}{2} m_p(a(t) + \mu g)R_B + \frac{J_m}{k}\frac{a(t)}{R_B} + \frac{C_m}{k}\frac{v(t)}{R_B} + \frac{1}{2} m_p g R_B \sin\alpha \qquad (15)$$

Rated torque:

$$m_r(t) = \frac{1}{2} R_B m_p \mu g + \frac{C_m}{k}\frac{v(t)}{R_B} + \frac{1}{2} m_p g R_B \sin\alpha \qquad (16)$$

2/ Turning in Place

From Equations 2, 7, 10, 13, and 14:
Peak torque:

$$m_t(t) = \frac{1}{2}\left[\frac{1}{d_3}\left(I_{zz} + m_p d_2^2\right)\varepsilon(t) + \mu m_p g\left(\frac{d_1}{d_1 + d_2} + \frac{d_2}{d_3}\right)\right] R_B + \frac{J_m}{k}\frac{\varepsilon(t)d_3}{R_B} + \frac{C_m}{k}\frac{\omega(t)d_3}{R_B} + \frac{1}{2} m_p g R_B \sin\alpha \qquad (17)$$

The term $m_p g R_B \sin\alpha$ is valid only for the worst case when the cart, turning in place, yields an uphill tangential speed of the cart centre of mass. In Equation 17, $\varepsilon(t)$ and $\omega(t)$ may be obtained from the a(t) and v(t) requirements developed by each drive motor, where the subscript twh refers to the tangential component with respect to the wheel:

$$\varepsilon(t) = a_{twh}(t)/d_3$$

$$\omega(t) = v_{twh}(t)/d_3 \qquad (17a)$$

Rated torque:

$$m_r(t) = \frac{1}{2}\mu m_p g\left(\frac{d_1}{d_1 + d_2} + \frac{d_2}{d_3}\right)R_B + \frac{C_m}{k}\frac{\omega(t)d_3}{R_B} + \frac{1}{2} m_p g R_B \sin\alpha \qquad (18)$$

3/ Rotation About One of the Driving Wheels

One of the driving motors is fully engaged in turning the cart if the other one is at rest. This situation corresponds to the cart rotation about one of its driving wheels and represents the most interesting case for the general cart motion in terms of maximum drive motor torques (peak and rated). From Equations 4, 9, 10, 13, and 14:

Peak torque:

$$m_t(t) = \frac{1}{2d_3}\left\{\left[\sum I_{zz} + m_p(d_3^2 + d_2^2)\right]\varepsilon(t) + \mu m_p g + \frac{d_1 d_3 + d_2\sqrt{d_3^2 + (d_1 + d_2)^2}}{d_1 + d_2}\right\} R_B$$

$$+ \frac{J_m}{k}\frac{\varepsilon(t)d_3}{R_B} + \frac{C_m}{k}\frac{\omega(t)d_3}{R_B} + \frac{1}{2} m_p g R_B \sin\alpha \qquad (19)$$

The term $m_p g R_B \sin\alpha$ exists only for the worst case, when the cart in its turning about one of the driving wheels, yields an uphill tangential speed of the cart centre of mass. In Equation 19, $\varepsilon(t)$ and $\omega(t)$ may be obtained from the $a(t)$ and $v(t)$ requirements developed by the specific drive motor:

$$\varepsilon(t) = \frac{a_{twh}(t)}{2d_3}$$

$$\omega(t) = \frac{v_{twh}(t)}{2d_3} \qquad (19a)$$

From Equations 17a and 19a, it is concluded that one motor turns the cart about one of the driving wheels (the other motor is at rest) twice as slow as two driving motors turning the cart about the point P, if the speed and acceleration of the motors is the same magnitude in both cases. Thus, to avoid an eventual lateral slip of the driving wheel (in this case, the turn is considered "sharp"), a motor acceleration cut by half is recommended.

Rated torque:

$$m_r(t) = \frac{1}{2d_3}\mu m_p g \frac{d_1 d_3 + d_2\sqrt{d_3^2 + (d_1 + d_2)^2}}{d_1 + d_2} R_B + \frac{C_m}{k}\frac{\omega(t) d_3}{R_B} + \frac{1}{2} m_p g R_B \sin\alpha \qquad (20)$$

3.5/ The Impact of the Cart Kinematic Requirements on the Drive Motors

Even for a relatively small cart linear speed for the task of cleaning, which is assumed to be 30 cm/s, it is a good practice to have a higher motor speed to test the cleaning performances of the prototype. Such an analysis may lead to the choice of the optimum operating speeds and accelerations/decelerations of the cart, which may optimize the total time necessary for the cleaning of a subway car, using the cleaning robot. Therefore, the proposed maximum slew speed of the robot is 0.5 m/s. However, for the prototype, the trajectory planning algorithm will be based on the slew speed of 30 cm/s. The linear acceleration of the cart is such that the cart slew speed is reached in 0.5 s. The acceleration does not greatly influence the cleaning time, but it dictates the trajectory error correction time, and its impact is significant on the peak torque of the drive motors. Thus, a reasonable acceleration requirement may be established iteratively, taking the available motors on the market into consideration. The angular acceleration and the angular speed of the cart in turning comes from the requirement that the performances of the drive motors should be those obtained from linear motion of the cart. Thus, $\varepsilon(t)$ and $\omega(t)$ of the cart are calculated from Equations 6 to 9, as follows:

1/ Straight-Line Motion

The terms $v(t)$ and $a(t)$ are based on the cleaning performances of the robot.

2/ Turning in Place

The terms $\dot{\theta}(t)$ and $\ddot{\theta}(t)$ are based on the kinematic requirements of the straight-line motion:

$$\omega(t) = \frac{R_B}{d_3}\dot{\theta}(t) = \frac{R_B}{d_3}\frac{v(t)}{R_B} = \frac{v(t)}{d_3}$$

$$\varepsilon(t) = \frac{a(t)}{d_3} \tag{21}$$

3/ Turning About One of the Driving Wheels

$$\omega(t) = \frac{v(t)}{2d_3}$$

$$\varepsilon(t) = \frac{a(t)}{2d_3} \tag{22}$$

These equations are the bridge between the drive motors and the cart trajectory planning strategy.

3.6/ Parameters Necessary for the Calculation of the Drive Motors

3.6.1/ Cart Moment of Inertia About Its Centre of Mass

The cart centre of mass is located at the intersection of the two planes of symmetry of the square cart (Figure 6). Thus, the cart moment of inertia about its centre of mass is

$$\circledcirc I_{zz} = \int_{body} (x^2 + y^2)dm = \frac{m_p}{12}(a^2 + b^2)$$

where

$a = b = 11$ in $= 0.2794$ m
$m_p = 200$ kg

Thus,

$$\circledcirc I_{zz} = 2.6021 \text{ kg}\cdot\text{m}^2$$

3.6.2/ Motor Armature Polar Moment of Inertia, J, and Viscous Coefficient of Friction, C

From PMI Motors' specifications of motors similar to those used in this application, the intrinsic motor constants of interest are

$J < 0.06$ kg·m^2
$C < 0.06$ Nm/krpm

These values lead to torques less than 0.01 Nm, so they may be ignored.

3.6.3/ Angle of Inclination α

The biggest angle α is obtained when the car floor is higher than the outside platform, such that the mobile cart must climb an inclined plate (laid over the adjacent surfaces) when it enters the car. If for each centimetre in height the lower end of the plate lies at 100 cm off the subway car door, then $\sin\alpha \approx tg\alpha = 0.01$. The resulting motor torque necessary for the cart to climb is in the same range as the torque due to rolling friction, since μ_{roll} and $\sin\alpha$ have very close values (compare Equations 16, 18, and 20).

3.6.4/ Geometrical Parameters

From Figures 6 and 7:

$d_1 = 0.17$ m (7 in)
$d_2 = 0.17$ m (7 in)
$d_3 = 0.228$ m (9 in)
$R_B = 0.08255$ m (3.25 in)

From the "Handbook for Applied Engineering Services" [2], the rolling coefficient of friction is chosen from the table on page 620:

$\mu = 0.01$ (rolling tire on a flat surface)

3.6.5/ Kinematic Parameters

1/ Linear Motion

It is assumed that

$v = 0.5$ m/s
$a = 1$ m/s^2

2/ Rotation in Place

From Equation 21:

$$\omega = \frac{0.5}{0.2286} = 2.187 \text{ rad/s}$$

$$\varepsilon = \frac{1}{0.2286} = 4.374 \text{ rad/s}^2$$

3/ Rotation About One of the Driving Wheels

The values of ω and ε are half of those for the rotation in place.

For all cases, the corresponding required speed and acceleration of the wheel shaft (which is equal to that of the motor shaft after the gear reduction) is obtained from Equations 6 and 7:

$$\dot{\theta} = 0.5/0.075 = 6.66 \text{ rad/s} = \frac{6.66}{2\pi} 60 \text{ rpm} = 63.69 \text{ rpm}$$

$$\ddot{\theta} = 1/0.075 = 13.33 \text{ rad/s}^2$$

These values for $\dot{\theta}$ and $\ddot{\theta}$ are based on the maximum speed and acceleration of the cart.

3.7/ Calculation of the Motor Torques

Based on previous discussion, the simplified formulae for calculations of the motor torques are obtained from Equations 15 + 20:

1/ Peak Torque

- Straight-Line Motion:

$$m_{t_1} = \frac{1}{2} m_p \left[a(t) + \mu g + g \sin\alpha\right] R_B \qquad (23)$$

- Turning in Place:

$$m_{t_2} = \frac{1}{2}\left[\frac{1}{d_3}\left(\circledcirc I_{zz} + m_p d_2^2\right)\frac{a(t)}{d_3} + \mu m_p g\left(\frac{d_1}{d_1 + d_2} + \frac{d_2}{d_3}\right)\right] R_B + \frac{1}{2} m_p g R_B \sin\alpha \qquad (24)$$

- Rotation About a Driving Wheel:

$$m_{t_3} = \frac{1}{2d_3}\left\{\left[\circledcirc I_{zz} + \mu_p(d_3^2 + d_2^2)\right]\frac{a(t)}{2d_3} + \mu m_p g \frac{d_1 d_3 + d_2\sqrt{d_3^2 + (d_1 + d_2)^2}}{d_1 + d_2}\right\} R_B + \frac{1}{2} m_p g R_B \sin\alpha \qquad (25)$$

2/ Rated Torque

- Straight-Line Motion:

$$m_{r_1} = \frac{1}{2} m_p g(\mu + \sin\alpha) R_B \qquad (26)$$

- Turning in Place:

$$m_{r_2} = \frac{1}{2} m_p g \left[\mu\left(\frac{d_1}{d_1 + d_2} + \frac{d_2}{d_3}\right) + \sin\alpha \right] R_B \qquad (27)$$

- Rotation About a Driving Wheel:

$$m_{r_3} = \frac{1}{2} m_p g \left[\frac{\mu}{d_3} \frac{d_1 d_3 + d_2 \sqrt{d_3^2 + (d_1 + d_2)^2}}{d_1 + d_2} + \sin\alpha \right] R_B \qquad (28)$$

The motor torques for this cart are calculated as follows:

m_{t_1} = 9.868 Nm
m_{t_2} = 8.887 Nm
m_{t_3} = 9.608 Nm

m_{r_1} = 1.618 Nm
m_{r_2} = 1.843 Nm
m_{r_3} = 1.962 Nm

3.8/ Motor Choice

The motor peak torque and the motor rated torque are chosen from the maximum values calculated in Section 3.7:

m_t = 9.868 Nm (87.49 in-lb)

m_r = 1.962 Nm (17.39 in-lb)

From the previous calculations, the motor rated speed after reduction is

$\dot{\theta}$ = 63.69 rpm

Among the PMI Motors' ferrite series gearmotors, a suitable motor for the mobile cart, under the specified kinematic, dynamic, and friction (working conditions) requirements, is the U9FGHD dc gearmotor with the gear ratio k = 79.2:1.

The obtained results of the analysis are shown in Tables 3 and 4.

The following are important facts about the selected motor:

- Because of the platform inaccuracies, the backlash of the gearbox is not the only factor which causes the absolute trajectory error of the cart. The exact total trajectory error may be found during the prototype testing.

- The encoder should be attached to the motor shaft.

- The required amplifier for the chosen motor is an AMP-10/70 A/V with a ±10 V input signal (cart power out is 22 W). Such an amplifier does not affect the stability of the DMC-200 controller, and there will be no need to readjust the controller filter. Even for such a case, the DMC-200 controller provides the user with compensation instructions. The values entered in the instructions are easily obtained analytically (based on the crossover frequency and phase margin) when all the system parameters (motor and load parameters) are known. The experimental method is useful when the motor and load parameters of the cart are unknown.

- For the vacuum tool, the motor used for the driving wheels is recommended:
  - The linear speeds of the tool and the driving wheels are close.
  - The torque required for the tool is smaller (the tool inertia is smaller than the cart inertia).
  - The amplifier and controller are the same as those for the driving wheels.

- The motor speed can be set at any value from zero to the rated speed by changing the amplifier input voltage.

3.9/ Encoder Choice

There are two limitations on the encoder choice:

1/ With the use of the DMC-200 controller, the maximum frequency of the encoder is limited to 62.5 kHz.

2/ The rated speed of the motor shaft (before the reduction) is 38 rpm x 79.2 = 3000 rpm (79.2 is the gear ratio).

If N is the encoder resolution (pulses per revolution), ν the frequency (pulses/s), and ω the rated speed of the motor shaft (rad/s), then

$$\nu = \frac{1}{2\pi} N\omega$$

or

$$N = \frac{2\pi\nu}{\omega}$$

Because of the expected encoder state width error (standard deviation of the overall encoder error) and controller sampling interval, ν_{max} is 62.5 kHz. With ω = 3000 rpm = 50 rps = 0.75 rad/s, N_{max} is estimated as

$$N_{max} = \frac{2 \times 3.14 \times 62\ 500}{314} = 1250 \text{ pulses (cycles)/rev}$$

For the DMC-200 controller, the encoder should be a two-channel quadrature with an optional index channel and a transistor-to-transistor logic (TTL) level output. The index channel is optional because the defined position (DP) command [1] sets the current motor position to a user-specified value (which may be zero); therefore, the encoder index channel is redundant.

A two-channel quadrature encoder with 500 to 1250 pulses/rev, no index channel, and a TTL level output (or any other good substitution) is recommended. Such an encoder has

$4N$ quadrature rounds/rev

or

$\frac{4N}{2\pi}$ quadrature counts/rad

The lower resolution limit is associated with the trajectory error value. Since the trajectory error is expected to be bigger because of cart geometrical inaccuracies, the only limit to be strictly obeyed is the higher resolution limit. If the higher resolution limit of 1250 pulses/rev is violated, the controller DMC-200 does not sample it properly (this means that the controller loses encoder data if the encoder resolution is higher than the resolution of the data transfer to the controller).

4/ MOTION CONTROL STRATEGY

4.1/ Cleaning Robot Configuration

The cleaning robot is assumed to be equipped as follows:

- two motors actuating two driving wheels;
- two motors actuating two cleaning tools;
- two DMC-200 controllers and amplifiers;
- four micro-switches (two pairs) mounted at the ends of the two tools;
- 10 micro-switches for emergency hazard: two pairs in front of the cart (left/right micro-switches for two tools), two at the rear platform, and two (fore/back) for each side of the platform;
- six ultrasonic sensors under on/off command of the host computer;
- guidance interface circuits (can be a microcomputer);
- an emergency light and emergency siren;
- indicator lamps;
- one start/stop push button and one reset button;
- a propane motor;
- a main power supply unit;
- an emergency power supply unit;
- two brakes normally applied under the on/off command of the host computer;
- six limit edges for the two cleaning tools (left/middle/right positions);
- two stoppers (brakes) and two dampers for the two tools;
- latched circuits, interlock circuits, buffers, and interface circuits;
- a host computer with I/O peripheral, including a driver for floppy disk or a bubble memory block.

4.2/ Robot Functions

Figure 3 shows the logical flow of information necessary to command and control the robot motion during the work.

To assess the nature and amount of information that is required by the robot software, a brief description of the modules (Figure 3) is given as follows:

- Trajectory planning converts user's input (robot path description and motion of the tools) into a table (Table 1) which consists of trajectory segments necessary to accomplish the required motion: mobile cart motion is described by its location (position and orientation), speed (linear and angular velocity), and acceleration/deceleration; tool motion is described by position with regard to a specific mobile robot location. Checkpoints should be defined for the guidance system.

At a checkpoint, the host computer checks the absolute location of the cart, using the information provided by the guidance system.

- Inverse kinematics converts Table 1 into Table 2, which consists of commands for the robot motors. Command may be expressed in encoder quadrature counts for the two driving motors and in homing (middle) and edge limits (right/left) position for the motors actuating the tools.

  Trajectory planning and inverse kinematics may be implemented off-line the host computer in a high-level language, and only Table 2 should be transferred (by a floppy disk) to the robot host computer. Another possible solution is the use of a bubble memory block rather than a floppy disk.

  For each subway car and cleaning strategy, a new Table 2 should be created.

- The guidance system measures the current location of the mobile cart with respect to the subway car, using six ultrasonic sensors. The guidance system works as follows. Each of the six sensors may be turned on/off by the host computer. For some specific positions of the mobile robot with respect to the configuration of the subway car, some of the sensors may interfere with each other and give false readings. The readings of the sensors are stored in some local buffers. The host computer takes data from the guidance buffers only when the monitoring program, the mobile robot brain (MRB), asks (trajectory checkpoint is encountered) for data. Data consist of measured distances and integers associated with the sensors that performed the measurements. Data may be transmitted to the host computer using a programmable adapter or a small microcomputer. The host computer compares data obtained from the guidance buffers with some expected values obtained from Table 2; then, it is decided whether corrective action is needed.

  Such an approach eliminates the need to know the source of the eventual trajectory errors.

  At the software level, when a trajectory checkpoint is encountered, a read statement is issued to take data from the guidance buffers.

  Direct kinematics is not needed since for each command the error of any of the robot motors may be obtained from the controller, DMC-200. The total accumulated error due to the control system is the algebraic sum of the errors associated with the accomplished trajectory segments. If reasonably small, such an accumulated error is easily identified by the guidance module and handled by the trajectory error correction module.

For the cleaning tools, the accumulation of errors is eliminated by the limit edge.

- The trajectory error correction module establishes a strategy of trajectory error handling, which issues adequate commands to the controllers. The idea is to bring the robot to a trajectory checkpoint from which the robot may continue its useful task.

  Because the guidance sensors cannot provide accurate measurements for an arbitrary platform position and orientation, the trajectory error correction module cannot have the absolute robot position, and the robot can get lost. A good strategy is to allow three trials of the trajectory error module, and if the algorithm fails, then an emergency is signalled. Thus, with more reliable and versatile sensors, a better trajectory error correction model may be developed and better robot autonomy achieved.

Guidance and trajectory error correction modules are used in the robot operating system, MRB; MRB monitors processes and interruptions. It faces three situations:
1/ It interacts with the user in local mode.
2/ It interacts with Table 2 in autonomous mode.
3/ The robot is in the dormant mode in emergency situation.

During its operation, MRB performs the following tasks:
- handles run-time routines to convert Table 2 into commands suitable for the DMC-200 controller (commands are down-loaded in ASCII characters one at a time with a carriage character after each command; in this way, the controller is programmed "on the fly");
- handles subroutines for guidance and trajectory error correction;
- reads status of the brakes, power supply units, and micro-switches (if any faulty condition occurs, stores the current robot parameters into a stack, then takes corrective action);
- if necessary, turns on emergency lights and a sound signal.

The local mode is useful to bring the robot to the homing position or to allow the user to take corrective action when the guidance and trajectory error modules are not able to bring the robot to a known position.

Besides some scientific operations, MRB is a decision maker, and it basically compares and branches information obtained from the following sources:
- user in the local mode;
- controllers;
- subroutines;

- buffers, which contain the status of robot subsystems, such as brakes, micro-switches, power supply units, interlock circuits (when in the dormant mode, the only activity of MRB is to check the interlock circuit status: set/reset);
- Table 2 in autonomous mode.

Brakes and other robot subsystems may be actuated by MRB through latched circuits.

4.3/ Theory of Operation

4.3.1/ Hardware Organization

The mobile robot is provided with three power supply units:
1/ a propane motor for the vacuum pump;
2/ a main battery unit for electrical motors, control circuits, a guidance system, and a host computer;
3/ a secondary battery unit for the emergency circuit.

The emergency power unit should be apart from the main power unit, since the robot should be able to signal an emergency if an electrical power failure of the main power unit occurs during the operation. The logical layout of the robot circuits is shown in Figure 2.

The START/STOP PUSH BUTTON turns on/off both main and emergency cir cuits. Thus, if this button is off, the emergency may not be signalled.

The solid-state relay (SSR) is normally open. Two conditions may cause the interlock circuit to close the SSR: power failure in one of the main circuit subsystems or a catastrophic trajectory error released by the host computer. For each interlock condition, an appropriate indicator lamp is lit and the emergency light and siren are activated. A description of the indicator lamps follows:
- L1 -- host computer or I/O power failure
- L2 -- electrical motor power failure
- L3 -- control circuits (including brakes) power failure
- L4 -- guidance system power failure
- L5 -- trajectory emergency

When the emergency condition is signalled, the RESET BUTTON is pushed and the emergency is removed. The robot command is taken over by the software, which displays instructions on the I/O screen of the monitor, if L1 is not the cause of the emergency. If L1 is the reason for the emergency, then all data from the main computer memory are lost.

If the START/STOP PUSH BUTTON is turned off, the interlock circuit is reset automatically.

4.3.2/ Robot Operation

In addition to the robot functions defined in Section 4.2, new functional requirements are defined to make the robot fully operative. They are implemented through a series of menus, which appear on the monitor screen and inform the user in a friendly way about actions he may take with respect to a predetermined user's goal. When the user turns the START/STOP PUSH BUTTON on, the computer responds with

```
MAIN MENU
RME   -- RESET MEMORY
RUN   -- RUN THE ROBOT
LOCAL -- LOCAL MODE

ENTER:
```

The cursor stays at ENTER and waits for the operator's answer. If none of these options is entered, the computer ignores the answer and keeps waiting (the conception is that the unexpected user's answer is always ignored by the computer program).

The RME command resets the computer memory. That means the program completes its session and all data from the main memory are lost. This command is especially useful when an emergency situation cannot be handled by the operator and he/she wants to restart the robot from the very beginning of the trajectory. If the RME command is applied after the START/STOP PUSH BUTTON is turned on, then RME is redundant and will be ignored.

The RUN command runs the robot, which means to put it in the autonomous mode in which the cleaning task is performed. When entered, the RUN command causes the computer to respond

```
ARE YOU SURE THAT THE ROBOT IS AT HOMING POSITION?  Y/N
ENTER: -
```

If Y is entered, the computer responds READY, waits 10 s, and starts the robot moving. On the screen, the number of the trajectory sequence being executed appears. If N is entered, the computer responds

```
DO YOU WANT TO USE LOCAL MODE?  Y/N
ENTER: -
```

If N is entered, the computer responds by displaying the Main Menu.
If Y is entered, then the computer responds with

```
LOCAL MODE MENU
ENTER DRIVING MOTORS DISPLACEMENTS (CM):
LEFT MOTOR: ------
RIGHT MOTOR: ------

ENTER POSITION OF TOOLS (LEFT/MIDDLE/RIGHT]:
MAIN TOOL (L/M/R):  ----
SECONDARY TOOL (L/M/R):  ----

TO EXIT, PRESS CTRL X
```

In the Local Mode Menu, the cursor is handled by ↑ ↓ → ← keys. When CTRL X is depressed, the computer responds with

```
IF YOU WANT TO START THE COMMANDED MOTION, PRESS CTRL P.
```

Now, the computer waits for CTRL P command and anything else entered is ignored. When CTRL P is pressed, computer responds with READY, waits 10 s, then starts the robot moving.

After the movement is completed, the computer displays

```
NEW COMMANDS? Y/N
ENTER: -
```

If Y is entered, computer responds displaying the Local Mode Menu.
If N is entered, computer responds displaying the Main Menu.
If an emergency (other than computer power failure) occurs at any time, the computer responds by the following message:

```
EMERGENCY SITUATION AT THE TRAJECTORY SEQUENCE IS q = ---:
REMOVE EMERGENCY
PRESS RESET BUTTON
```

When the emergency is removed and the interlock is reset by pressing the RESET BUTTON, the host computer is informed by the interlock circuit that the emergency is over, and as a result of that the main menu is displayed on the screen. Before the emergency situation is signalled, all the current information is stored in a stack in the computer main memory.

After the emergency is removed, the operator may look at the robot current position and decide the following:

- The robot may continue the trajectory immediately because the robot did not lose the trajectory. The operator may enter
  - RUN, press RETURN key and wait for the answer;
  - then Y, press RETURN key. In response, the robot will continue its movement.

- The robot may not continue the trajectory immediately because the robot lost the trajectory. If the operator recognizes the starting point of the trajectory sequence being interrupted, he/she may
  - enter LOCAL, press RETURN key, and wait for the answer;
  - then obey instructions on the screen (robot is returned by the operator at the beginning of the previous trajectory sequence).

- The robot may not continue the trajectory at all. The operator may enter
  - RME, press RETURN key, and wait for answer;
  - then obey instructions on the screen (robot is returned to the homing position or taken away).

If the START/STOP PUSH BUTTON is turned off during the robot work, brakes will automatically be applied (since they are normally applied), and the content of the computer main memory is lost. There is no provision for better handling of this situation.

The operation mode which is described in this chapter should be understood by the operator. It fully depicts the robot behaviour; the outcome of any attempt to use the robot differently than previously defined is not guaranteed.

4.4/ Software Architecture

Based on the robot theory of operation, the software architecture represents the robot brain and all its connections to the surrounding equipment. With respect to the logical flow of information (Figure 3), part of the data may be processed off-line (trajectory planning and inverse kinematics modules) and the remaining data on-line. The table of commands which are produced by the off-line program comes to the main computer on a floppy disk (or on another media) as an input. For a new environment associated with a cleaning task, a new floppy disk may be used. Such a software organization permits the creating and storing of a versatile library, leaving the robot operating system unchanged. The organization of the robot operation system (software architecture) is

shown in Figure 4.

The architecture of the off-line computer program may be produced, based on the trajectory planning and inverse kinematics modules.

The communication between the host computer and peripheral equipment is accomplished by using a programmable interface adapter (PIA) or a reasonable substitution of it. The robot operating system consists of a master program; mobile robot brain (MRB), which is primarily a decision maker; and a series of utility programs that basically execute decisions. Run-time routines take care of I/O operation, based on priorities in the robot information system. The need for a priority comes from the communication requirement: who should open a talk for the conversation between two modules? This question is uniquely solved by the new question: why have a talk between two modules? The answer to this question is found in the definition of the robot functions needed to perform some predetermined tasks. The priorities are assigned as follows:

- 0 - interruption from power supply units (computer power failure is not included)
- 1 - interruption from emergency micro-switches
- 2 - interruption from controllers DMC-200
- 3 - interruption from brakes
- 4 - interruption from I/O device
- 5 - interruption from guidance system

Routine jobs, high-priority real-time jobs, or special jobs should all be performed at the same time. A technique for handling numerous routines or programs simultaneously by overlapping or interleaving their execution is called "multiprogramming." Priority multiprogramming is oriented toward concurrent operation of several types of applications. For example, the first priority level can be reserved for a program that must provide rapid responses to real-time devices. Inside this real-time program, a scheduling plan or formula is established. Such a schedule is based on the list of priorities. Interruptions may be handled using "an interrupt vectoring": polling sequence services interrupt requests, such that the value of each interruption request can be used as a system address to transfer control for the appropriate response routine. Details should be solved at a later time.

The robot operating system should be understood as an executable command (EC) file developed and executed in the computer operating system which accompanies the host computer. In such a case, the RMB is composed of micro-instructions (commands) in the EC file, utility programs are implemented in a high-level language (FORTRAN 77, C, or Forth), and run-time routines are coded in Assembler. With this conception, only one program

resides in the main memory at a given time.

The host computer and computer operating system should meet the following requirements:

- suitable for real-time applications
- have a straightforward EC capability
- support one of the languages: F77, C, Forth
- small dimensions
- low cost

The computer memory and processing speed requirements are based on the number of instructions and data which have to reside in the main and auxiliary memories, as well as on the number of interruptions to be handled. These requirements may be known when a mathematical model and a pseudocode of the utility programs are established.

Figure 4 shows the macro-functions of the robot operating system rather than cumbersome details.

When the computer is turned on, a "boot" procedure initializes the computer operating system. When this procedure is completed, the operating system automatically loads file MRB into the main memory and initiates execution. In MRB, the highest level of the robot operating system is stored. It consists of subordinated executable files that are programmed inside the MRB file using a set of macro-instructions recognized by the computer operating system. The structure of the MRB file is shown in Figure 5.

Each subordinated executable file has a structure similar to that of the MRB file. A simplified block diagram of the level 2 is not suitable at this project stage. Such a scheme should be built starting with the base of the programming pyramid of the RUN and LOCAL modules.

The third programming level is written in a high-level language. The third level communicates only with the second programming level. The programs belonging to the third level call on execution low-level subroutines, which are written in a high-level language or Assembler language.

5/ MOTION CONTROL -- TRAJECTORY PLANNING MODULE

5.1/ Introduction

The trajectory planning module translates a set of user's information containing the structured environment (working space) description and the description of the task to be performed (the motion of the tools) into a table (Table 2) that comprises the motion characteristics to be commanded (through DMC-200 controllers) to the motors. The trajectory planning task is accomplished in two steps:

1/ User's input (description of the working envelope) is translated into a table (Table 1) containing the description of the trajectory segments of the cart point P (midpoint between the cart driving wheels) and the tool points P_1 and P_2 (P_1 is the midpoint of the main tool and P_2 is the midpoint of the secondary tool).

2/ Table 1 is translated into Table 2, which contains commands to the motors. These commands should be useful for the jog mode of the DMC-200 controller. The set of equations which transforms Table 1 into Table 2 is called inverse kinematics (or inverse kinematic equations) of the robotic cart.

5.2/ Trajectory Steps and Pseudocode

5.2.1/ Trajectory Steps

The subway car configuration is shown in Figure 11. Dimensions are given in millimetres. The cart path with the trajectory steps and checkpoints is shown in Figure 12. Two types of the trajectory segments of motion are used: translation of the cart point P and cart rotation about point P (turning on spot). These types of motion segments are sufficient to fully describe the path of the cart. The integration of the cart and tool motion consists of the adjustment of the position of the tools during the cart motion such that no impact between the cleaning tools and the subway car (seats, walls, etc.) occurs while the cleaning is done. The integration is discussed in Section 5.4.

The robotic cart starts its motion from the homing co-ordinate system $O_H x_H y_H$ (Figure 12). The motion (kinematic characteristics) of the cart is represented by the location L, speed S, and acceleration A of the point P (midpoint between driving wheels) of the robotic cart. The following definitions are introduced (Figure 13):

(x,y) cart position: co-ordinates of the point P

θ cart orientation: angle between the cart longitudinal axis of symmetry and co-ordinate axis x

v cart linear velocity: linear velocity of the point P along the cart longitudinal axis of symmetry

ω cart angular velocity: rotation of the cart about point P

a cart linear acceleration: linear acceleration of the point P along the cart longitudinal axis of symmetry

ε cart angular acceleration: angular acceleration of the cart $(d\omega/dt)$

Therefore:

$$\begin{aligned} L &= (x,y;\theta) \\ S &= (v,\omega) \\ A &= (a,\varepsilon) \end{aligned} \qquad (1)$$

The motion of the cart in the subway car is related to the working co-ordinate system $O_E x_E y_E$ which is attached to the subway car (Figure 12). The working co-ordinate system is fixed with respect to the homing co-ordinate system; thus,

$$\begin{aligned} x_{OE} &= L_1 \text{ (mm)} \\ y_{OE} &= L_2 \text{ (mm)} \end{aligned} \qquad (2)$$

where L_1 and L_2 should be determined at a later stage.

Any eventual inaccuracy of the position of the point O_E with respect to the homing co-ordinate system should be handled by the trajectory error correction module. The trajectory steps are shown in Figure 12. They are planned such that the path of the cart is minimized.

5.2.2/ Pseudocode of the Trajectory

The path of the robotic cart consists of 49 trajectory sequences (43 translations and six rotations in place, e.g., about the point P). At the end of each trajectory sequence the robotic cart is at rest because a new motion type or some tool manipulations have to be performed. Each trajectory sequence consists of a certain number of trajectory segments. After a trajectory segment is accomplished (travelled), the guidance

system checks the absolute cart position while the platform is in motion. Thus, the division of each trajectory sequence into trajectory segments is imposed by the trajectory checkpoints. The controllers are programmed in the jog mode "on the fly." When a trajectory segment is accomplished, the next instruction to the controller commands the subsequent trajectory step, depending on the absolute cart position with respect to the planned one. The trajectory steps are give in Table 1. In this Table, point $C_i, i=1,n$ is the sequence of checkpoints that is used by the host computer to control the distance sensors of the robotic cart guidance system.

5.2.3/ Kinematic Equations of the Robotic Cart

For the two types of trajectory segments of the robotic cart, the set of equations which leads to the motion characteristics of the shafts of the driving motors is obtained as follows:

- **Translational Motion:** L -- left motor
  R -- right motor

$$\theta_L = \theta_R = \frac{1}{k}\frac{x}{R} nt \ (rad)$$

$$\omega_L = \omega_R = \frac{1}{k}\frac{v}{R} nt \ (rad/s) \qquad (3)$$

$$\varepsilon_L = \varepsilon_R = \frac{1}{k}\frac{a}{R} nt \ (rad/s^2)$$

$$x = vt + \frac{1}{2} at^2 \ (m) \qquad (4)$$

Equation 4 is the solution of the differential equation $\ddot{x} = a$.

- **Rotation About Point P**

$$\mp\theta_L = \pm \theta_R = \frac{1}{k}\frac{\theta d_3}{R} nr \qquad (rad)$$

$$\mp\omega_L = \pm \omega_R = \frac{1}{k}\frac{\omega d_3}{R} nr \qquad (rad/s) \qquad (5)$$

$$\mp\varepsilon_L = \pm \varepsilon_R = \frac{1}{k}\frac{\varepsilon d_3}{R} nr \ (rad/s^2)$$

$$\theta = \omega t + \frac{1}{2} \varepsilon t^2 \qquad (6)$$

where:

Θ_L, Θ_R = left/right motor shaft position angle (rad)
ω_L, ω_R = left/right motor shaft angular velocity (rad/s)
ε_L, ε_R = left/right motor shaft angular acceleration (rad/s^2)
k = gear ratio defined as wheel-shaft angle/motor-shaft angle <1
R = driving wheel radius (mm)
d_3 = distance between point P and each of the driving wheels (mm)
t = time of the cart motion (s)
nt = flag for translation case: nt = 1 forward
nt = -1 reverse
nr = flag for rotation case about P: nr = 1 left turn
nr = -1 right turn
x = length of the translational segment (mm)
v = linear velocity of the cart (mm/s)
a = linear acceleration of the cart (mm/s^2)
Θ = cart orientation (rad)
ω = angular velocity of the cart (rad/s)
ε = angular acceleration of the cart (rad/s^2)

The terms x, v, a, Θ, ω, and ε are always positively defined because flags nt and nr regulate the sign of Θ_R, Θ_L, ω_R, ω_L, ε_L, and ε_R.

5.3/ Task Steps and Pseudocode

5.3.1/ Floor Coverage

It is reasonable to clean the subway car up to 12.7 mm (0.5 in) from the walls of the car (see the dashed contour in Figure 14). Observe that the floor coverage (cleaning of the car floor by two movable tools) depends on the relative motion between the cart and each tool respectively. Thus, the cart motion is defined first (see the motion profile of the cart in Figure 14).

The main tool may be commanded in the same way as the driving wheels, while the small, secondary tool has only three positions: extended right or left, and retracted.

The main tool may encounter the following situations:
- The tool is stationary during the cart motion.
- The tool moves while the cart is stationary.
- The tool moves with a constant speed while the cart is approaching a wall edge.
- The tool follows a convex circle while the cart is translating.
- The tool follows a concave circle while the cart is translating.

5.3.2/ Kinematic Equations of the Main Tool

As a mechanism, the main tool is a rack which is translated by a pinion of radius r = 38.1 mm (1.5 in) mounted on the motor shaft (after the motor gearbox). If θ_s, ω_s, and ε_s are the angle, angular velocity, and angular acceleration of the motor shaft, the position x_t, linear velocity v_t, and linear acceleration a_t of the tool are given as follows:

$$\begin{aligned} x_t &= \theta_s r \ (m) \\ v_t &= \omega_s r \ (m/s) \\ a_t &= \varepsilon_s r \ (m/s^2) \end{aligned} \tag{7}$$

where the angle is given in rad and r in mm.

From (7), the motor-shaft kinematic parameters are obtained as follows:

$$\begin{aligned} \theta_s &= x_t/r \ (rad) \\ \omega_s &= v_t/r \ (rad/s) \\ \varepsilon_s &= a_t/r \ (rad/s^2) \end{aligned} \tag{8}$$

The parameters x_t, v_t, and a_t should be defined such that the floor is completely covered (cleaned) by the main tool. The narrow strip along the longitudinal axis of the subway car is always covered by the small tool.

The most demanding job is the cleaning of the rounded corners of the subway car walls. The two typical situations are shown in Figures 15 and 16. The kinematic equations of the main tool for the concave circle case are obtained as follows:

1/ Cart acceleration is zero; cart speed is v_c = constant.
Then, the position, speed, and acceleration of the main tool are as follows:

$$\begin{aligned} x_t &= R_c - (R_c^2 - v_c^2 t^2)^{1/2} \\ v_t &= v_c \left(\frac{R_c^2}{v_c^2 t^2} - 1\right)^{-1/2} \\ a_t &= \frac{R_c^2}{v_c t^3} \left(\frac{R_c^2}{v_c^2 t^2} - 1\right)^{-3/2} \end{aligned} \tag{9}$$

Since the motion time t is the same for the cart and tool, t may be

obtained from Equation 5 for $a_c = o$ (cart acceleration is zero): $t = x_c(t)/v_c$. Near the tangential point T_2 (the end of the corner), v_t and a_t become very big (v_t, $a_t \rightarrow \infty$) since time t becomes $t = R_c/v_c$ and the last two equations in Equation 9 have a singularity, namely:

$$\frac{R_c^2}{v_c^2 t^2} - 1 \rightarrow 0$$

For the tool motion, there are two constraints:

a) The speed of the tool is limited by the maximum speed $v_{max,tool}$, which is the upper limit for a proper cleaning of the floor.

b) The acceleration of the tool is imposed by the motor peak torque T_{peak} and dynamic equations of the transmission and main tool system:

$$r\, m\, a_t + I_{trans}\, \varepsilon_s + c_{damping}\, \omega_s + J_p \varepsilon_s = T_{peak} \qquad (10)$$

where:

$$\varepsilon_s = d\omega_s/dt = \dot{\omega}_s$$

$$a = r\varepsilon_s$$

$c_{damping}$ = damping coefficient of the tool

J_p = motor polar moment of inertia $\simeq 0$

I_{trans} = polar moment of inertia of the transmission system

Thus:

$$\dot{\omega}_s (\, mr^2 + I_{trans} + J_p) + c_{damping}\, \omega_s = T_{peak} \qquad (11)$$

The solution of the linear differential, Equation 11, is:

For $c_{damping} \neq 0$:

$$\omega_s(t) = T_{peak}\, e^{-\left(\frac{c_{damping}}{mr^2 + I_{trans} + J_p}\right)} + \frac{T_{peak}}{c_{damping}} \qquad (12)$$

$$\varepsilon_s(t) = \dot{\omega}_s = - \frac{c_{damping}\, T_{peak}}{mr^2 + I_{trans} + J_p}\, e^{-\left(\frac{c_{damping}}{mr^2 + I_{trans} + J_p}\, t\right)} \qquad (13)$$

Thus:

$$\varepsilon_s(t)_{max} = \frac{T_{peak}\, c_{damping}}{mr^2 + I_{trans} + J_p} \qquad (14)$$

$$\omega_s(t)_{max} = T_{peak} \left(1 + \frac{1}{c_{damping}} \right) \qquad (15)$$

On the other hand, ω_s index is obtained from Equation 8:

$$\omega_{s\ max} = \frac{v_{max,tool}}{r} \qquad (16)$$

Thus, from Equations 15 and 16, $\omega_{s\ max}$ is obtained as

$$\omega_{s\ max} = \text{minimum} \left[T_{peak} \left(1 + \frac{1}{c_{damping}}\right), \frac{v_{max,tool}}{r} \right] \qquad (17)$$

and

$$\varepsilon_{s\ max} = \frac{T_{peak}\, c_{damping}}{mr^2 + I_{trans} + J_p} \qquad (18)$$

where

- J_p and T_{peak} are obtained from the motor characteristics;
- m, I_{trans}, and $c_{damping}$ are obtained from the transmission and main tool system;
- $v_{max,tool}$ is obtained from the practice of cleaning by using a specific vacuum cleaner.

Note that $v_{max,tool}$ also depends on how dirty the floor is. For the first iteration of the robotic cart testing, $v_{max,tool}$ has to be "guessed" and then later modified according to the cleaning tests.

2/ Cart acceleration is different than zero, but it is a constant function of time t (this is the case in the profile motion of the DMC-200 controller).

Then, the position, speed, and acceleration of the main tool are as follows:

a) Cart is accelerated; initial cart speed is $v_{c\ init} = 0$:

$$x_t = R_c - (R_c^2 - a_c^2 \frac{t^4}{4})^{1/2}$$

$$v_t = \frac{1}{2} a_c^2 t^3 (R_c^2 - a_c^2 \frac{t^4}{4})^{-1/2} \qquad (19)$$

$$a_t = \frac{1}{2} a_c^2 \left[3t^2 (R_c^2 - a_c^2 \frac{t^4}{4})^{-1/2} + \frac{1}{2} a_c^2 t^6 (R_c^2 - a_c^2 \frac{t^4}{4})^{-3/2} \right]$$

where $t = (2x_c/a_c)^{1/2}$

A singularity of Equation 19 is encountered if the point T_2 (the end of the corner) is reached during the acceleration time of the cart:

$$t = \sqrt{\frac{2R_c}{a_c}}$$

This case should be avoided in practice.

b) Cart is decelerated; initial cart speed is $v_{c\ init} = v_c \neq 0$ (this is the case in the cart profile motion):

$$x_t = R_c - (R_c^2 + a_c v_c t^3 - a_c^2 \frac{t^4}{4} - v_c^2 t^2)^{1/2}$$

$$v_t = (\frac{3}{2} t^2 a_c v_c - \frac{1}{2} a_c^2 t^3 - v_c^2 t) (R_c^2 + a_c v_c t^3 - a_c^2 \frac{t^4}{4} - v_c t^2)^{-1/2}$$

$$a_t = (-\frac{3}{2} a_c v_c t + \frac{3}{4} a_c^2 t^2 + \frac{1}{2} v_c^2) (3a_c v_c t^2 - a_c^2 t^3 - 2v_c t)$$
$$(R_c{}^2 + a_c v_c t^3 - a_c^2 c \frac{t^4}{4} - v_c t^2)^{-3/2} \qquad (20)$$

where:

$$t = (\frac{v_c^2}{a_c^2} + 2 \frac{x_c}{a_c})^{1/2} \frac{vc}{a_c}$$

A singularity of Equation 20 is encountered for

$$t = \left(\frac{v_c^2}{a_c^2} + 2\,\frac{R_c}{a_c}\right)^{1/2} - \frac{v_c}{a_c}$$

This case should avoided in practice.

The actual values for ε_s and ω_s are obtained by substituting Equation 20 into Equation 8. Their maximum values should meet the conditions given by the Equations 17 and 18.

The kinematic equations of main tool for the convex circle (Figure 16) are obtained as follows:

3/ Cart acceleration is zero; cart speed is v_c = constant. The position, speed, and acceleration of the main tool are as follows:

$$x_t = R_c - (2R_c v_c t - v_c^2 t^2)^{1/2}$$

$$v_t = (R_c v_c - v_c^2 t)\,(2R_c v_c t - v_c^2 t^2)^{-1/2}$$

$$a_t = -[v_c^2(2R_c v_c - v_c^2 t^2) + (R_c v_c - v_c^2 t)^2\,(2R_c v_c t - v_c^2 t^2)^{-3/2}] \qquad (21)$$

A singularity of Equation 21 is for time t = 0 (start of motion, point T1).

4/ Cart acceleration is constant.

a) Cart is accelerated; initial cart speed is $v_{c\ init} \neq 0$:

$$x_t = R_c - \left(2R_c v_c t + R_c a_c t^2 - v_c^2 t^2 - a_c^2\,\frac{t^4}{4}\right)^{1/2}$$

$$v_t = \left(R_c v_c - R_c a_c t - v_c^2 t - \frac{1}{2}\,a_c^2 t^3\right)$$

$$\left(2R_c v_c t + R_c a_c t^2 - a_c^2\,\frac{t^4}{4}\right)^{-1/2} \qquad (22)$$

$$a_t = \left(R_c a_c - v_c^2 - \frac{3}{2}\,a_c^2 t^2\right)\left(2R_c v_c t + R_c a_c t^2 - v_c^2 t^2 - a_c^2\,\frac{t^4}{4}\right)^{-1/2}$$

$$- \left(R_c v_c + R_c a_c t - v_c^2 t - \frac{1}{2}\,a_c^2 t^3\right)^2$$

$$(2R_c v_c t + R_c a_c t^2 - v_c^2 t^2 - a_c^2 \frac{t^4}{4})^{-3/2}$$

A singularity of Equation 22 is encountered for time t = 0 (start of motion, point T_1)

b) Cart is decelerated; initial cart speed is $v_{c\ init} \neq 0$ (this is the case in the cart profile motion).

$$x_t = R_c - (2R_c v_c t - R_c a_c t^2 - v_c^2 t^2 - a_c^2 \frac{t^4}{4})^{1/2}$$

$$v_t = (R_c v_c - R_c a_c t - v_c^2 t - \frac{1}{2} a_c^2 t^3)$$

$$(2R_c v_c t - R_c a_c t^2 - v_c{}^2 t^2 - a_c{}^2 \frac{t^4}{4})^{-1/2} \qquad (23)$$

$$a_t = -(R_c a_c - v_c{}^2 - \frac{3}{2} a_c^2 t^2)\ (2R_c v_c t + R_c a_c t^2 - v_c^2 t^2 - a_c{}^2 \frac{t^4}{4})^{-1/2}$$

$$-(R_c v_c - R_c a_c t - v_c{}^2 t - \frac{1}{2} a_c{}^2 t^3)^2$$

$$(2R_c v_c t - R_c a_c t^2 - v_c^2 t^2 - a_c^2 \frac{t^4}{4})^{-3/2}$$

A singularity of Equation 23 is obtained for t = 0 (start of the motion, point T_1).

Analysing the kinematic equations of the main tool motion, it is clear that they are not useful in the form they are given in the Equations 9, 19, 20, 21, 22, and 23. Driving motors have to be programmed with a constant acceleration/deceleration of the motor shaft to achieve a desired slew speed. On the other hand, the acceleration is limited by the motor peak torque, T_{peak} (see Equation 18). Since the tool motion is damped, the speed of tool has two limitations, given by the Equation 17. The engineering approach to tool kinematics requires a fine number of commands, thus a discretized form of the kinematic equations of the tool, and, at the same time, a cleaning strategy that will result in a proper cleaning of the corners. In such a case, the strategy of motion is to accelerate/decelerate the tool while the cart is decelerated/ accelerated, depending on the type of corner (convex/concave).

5.3.3/ Pseudocode of the Task Steps

The engineering solution of the circle case for the robotic cart/main tool motion is given here. The equations from Section 5.2 are continuous

functions of the time t. A discretization consists of establishing the motion profile for the cart and main tool for discrete time intervals.

For a concave-circle-type corner (Figure 17), the cart velocity profile is

$$x(t) = v_c t - a_c \frac{t^2}{2} \tag{24}$$

Since the corner arc C_3T_2 is very abrupt, the cart is brought to a full stop after the distance $d = R_c - \frac{R_c}{12} = \frac{11}{12} R_c$ is travelled. In such a case, the cart acceleration a_c and the time needed to stop the cart t are

$$a_c = \frac{v_c^2}{2 \frac{11}{12} R_c} = \frac{6}{11} \frac{V_c^2}{R_c} \tag{25}$$

$$t^* = \frac{2 \frac{11}{12} R_c}{v_c} = \frac{11}{6} \frac{R_c}{v_c} \tag{26}$$

From Equation 24, the time t as a function of the cart travelled distance x(t) is

$$t = \frac{11}{6} \frac{R_c}{v_c} \left[1 - \sqrt{1 - \frac{12}{11} \frac{x(t)}{R_c}}\right] \tag{27}$$

For $x(t) = \frac{R_c}{3}$, the time t_1 is

$$t_1 = \frac{R_c}{v_c} \frac{11}{6} \left(1 - \sqrt{\frac{7}{10}}\right) \tag{28}$$

For $x(t) = \frac{2R_c}{3}$, the time t_2 is

$$t_2 = \frac{R_c}{v_c} \frac{11}{6} \left[1 - \sqrt{\frac{3}{11}}\right] \tag{29}$$

For $x(t) = \frac{11R_c}{12}$, the time t_3 is

$$t_3 = \frac{R_c}{v_c}\frac{11}{6} \tag{30}$$

From the Equations 20 to 30, it is clear that

$$t_2 - t_1 < t_3 - t_2 \tag{31}$$

Indeed,

$$(1 - \sqrt{\frac{3}{11}}) - (1 - \sqrt{\frac{7}{11}}) < 1 - (1 - \sqrt{\frac{3}{11}})$$

Since

$$\sqrt{\frac{7}{11}} < 2\sqrt{\frac{3}{11}} = \sqrt{\frac{12}{11}}$$

Equation 31 shows that the time decreases parabolically for each travelled x(t) during the cleaning of the concave corner.

During this period of time, $t^* = \frac{11}{6}\frac{R_c}{v_c}$, the tool motion is governed by Equation 20. The tool speed $v_t(t)$ and tool accelerations $a_t(t)$ are highly non-linear functions of time, and the controller DMC-200 can handle only a uniformly accelerated/decelerated motion. The linearization of Equation 20 consists of assuming that the tool motion is uniformly accelerated for arcs S_1, S_2, S_3, and S_4. In such a case, the point A of the tool end follows an arc of curve which is slightly different than the arcs $S_i, i = 1,4$ of the circle describing the corner. But, at the end of each such approximate arc, the circle point $C_i, i = 1,4$ should be achieved. This condition is mathematically expressed as

$$y_1(t_1 - t_0) = v_{0t}(t_1 - t_0) + a_{1t}\frac{(t_1 - t_0)^2}{2} \tag{32}$$

$$y_2(t_2 - t_1) = v_{1t}(t_2 - t_1) + a_{2t}\frac{(t_2 - t_1)^2}{2} \tag{33}$$

$$y_3(t_3 - t_2) = V_{2t}(t_3 - t_2) + a_{3t}\frac{(t_3 - t_2)^2}{2} \tag{34}$$

where $t_0 = 0$ (initial time) and $v_{0t} = 0$ (initial tool velocity). The remaining velocities of the tool are

$$v_{1t} = a_{1t}(t_1 - t_0) + v_{0t} \tag{35}$$

$$v_{2t} = a_{2t}(t_2 - t_1) + v_{1t} \tag{36}$$

$$v_{3t} = a_{3t}(t_3 - t_2) + v_{2t} \tag{37}$$

The goal is to obtain a_{1t}, a_{2t}, and a_{3t}, which are useful for the controller. The velocity v_{3t} is important for the timing of the movement along S_4 (Figure 17). First, y_1, y_2, and y_3 are obtained from Figure 17 as follows:

$$y_1 = R_c - \sqrt{R_c^2 - \frac{R_c^2}{9}} = R_c\left(1 - \sqrt{\frac{8}{9}}\right) = 0.057R_c$$

$$y_2 = R_c - \sqrt{R_c^2 - \frac{4R_c^2}{9}} - y_1 = R_c\left(\sqrt{\frac{8}{9}} - \sqrt{\frac{5}{9}}\right) = 0.1974R_c$$

$$y_3 = R_c - \sqrt{R_{c2} - \frac{121}{144}R_c^2} - (y_1 + y_2) = R_c\left(\sqrt{\frac{5}{9}} - \sqrt{\frac{13}{144}}\right) = 0.4448R_c$$

$$y_4 = R_c - (y_1 + y_2 + y_3) = R_c\sqrt{\frac{13}{144}} = 0.3004R_c \tag{38}$$

From Equations 28 to 30:

$$t_1 - t_0 = 0.3708\,\frac{R_c}{v_c}$$

$$t_2 - t_1 = 0.505\,\frac{R_c}{v_c}$$

$$t_3 - t_2 = 0.963\,\frac{R_c}{v_c} \tag{39}$$

Thus, from (39) and (32), a_{1t} is obtained as

$$a_{1t} = \frac{2y_1}{(t_1 - t_0)^2} = 2\,\frac{0.057R_c}{(0.3708)^2\,\frac{R_c^2}{v_c^2}} = 0.829\,\frac{v_c^2}{R_c}$$

The speed v_{1t} is obtained from Equation 35 as follows:

$$v_{1t} = a_{1t}(t_1 - t_0) + v_{0t} = 0.829 + \frac{V_c^2}{R_c} 0.3708 \frac{R_c}{v_c} = 0.307\ v_c$$

Now, the acceleration a_{2t} is found as follows:

$$a_{2t} = \frac{y_2 - v_{1t}(t_2 - t_1)}{1/2(t_2 - t_1)^2} = \frac{0.1974R_c - 0.307v_c 0.505 \frac{RC}{V_c}}{1/2\ (0.505 \frac{R_c^2}{V_c})}$$

$$= 0.332 \frac{v_c^2}{R_c}$$

Then:

$$v_{2t} = a_{2t}(t_2 - t_1) + v_{1t} = 0.332 \frac{v_c}{R_c} (0.505 \frac{R_c}{v_c}^2) + 0.307v_c$$

$$= 0.0846v_c + 0.307v_c = 0.391v_c$$

Now, a_{3t} is

$$a_{3t} = \frac{y_3 - v_{2t}(t_3 - t_2)}{1/2(t_3 - t_2)^2} = \frac{0.445R_c - 0.391v_c 0.963 \frac{R_c}{V_c}}{1/2\ (0.963 \frac{R_c}{v_c})^2}$$

$$= 0.146 \frac{v_c^2}{R_c}$$

At the end of the movement along the last parabolic arc, the speed of the tool is

$$v_{3t} = a_{3t}(t_3 - t_2) + v_{2t} = 0.146 \frac{v_c^2}{R_c} 0.963 \frac{R_c}{v_c} + 0.391\ v_c$$

$$= 0.531v_c$$

Thus, for the concave corner, the time for cleaning is $t^* = \frac{11}{6}\frac{R_c}{v_c}$. With R_c = 500 mm and V_c = 1 ft/s = 304 mm/s, the cleaning time t^* is $t^* \approx 3.01$ s. During this time, the cart is decelerated with

$a_c = \frac{6}{11}\frac{v_c^2}{R_c} = 100.8$ mm/s^2 and brought to a full stop. The main tool is accelerated in three phases:

1/ $a_{1t} = 164.87$ mm/s^2

2/ $a_{2t} = 61.36$ mm/s^2

3/ $a_{3t} = 26.98$ mm/s^2

For the cart:

$$a_c = -100.8 \text{ mm/s}^2$$

The cart is kept at rest while the main tool is retracted along the segment S_4. Segment S_4 is calculated as follows:

$$S_4 = y_4 + d_0$$

Thus:

$$S_4 = 0.3004R_c + 0.00 = 150.2 \text{ mm}$$

Along the segment S_4, the speed of the tool is kept constant, $v_{3t} = 0.531v_c = 161.42$ mm/s, such that the cart is at rest for

$$t_{rest} = \frac{S_4}{v_{3t}} = 0.930 \text{ s}$$

At this point, the task of corner cleaning is finished, with a total cleaning time of

$$t_{total} = t^* + t_{rest} = 3.01 + 0.930 = 3.940 \text{ s}$$

Note that all commanded accelerations and speeds (for cart and main tool) are kept reasonably low, such that the quality of cleaning does not deteriorate. On the other hand, the obtained engineering solution is not optimal, because the cart is brought to a full stop in one step and the resulting accelerations of the tool are low and decreasing. The concave

corner is divided into four segments. A finer division of the corner leads to a more precise tool trajectory. Both inconveniences may be removed by better trajectory planning for the corner. This requires a computer program because the number of calculations is large. Once again, such a computer program would have the purpose of cleaning time minimization and a precise following of the corner contour by the tool.

For the trajectory planning of the convex corner, an analogous approach should be used. In the subway car, there is only one convex corner, and it can be ignored.

For the situation when a corner is cleaned starting from the upper (ending) point, the described procedure is still valid, with the only difference that the sequence of equations of the tool and commands to the controllers should be reversed.

5.4/ Integration

As described in Sections 5.1 and 5.2.1, integration shows the synchronization of motion of the robotic cart, main tool, and secondary tool. If motors 1, 2, 3, and 4 move the left driving wheel, right driving wheel, main tool, and secondary tool, respectively, then all the motors are simultaneously commanded through DMC-200 controllers in the jog mode. This situation is shown in the Table 2. The kinematic equations of the cart and main tool are used to calculate the shaft angle, angular speed, and acceleration of each motor, expressed in encoder counts, encoder counts/s, and encoder counts/s^2, respectively.

5.5/ Motion Control Strategy Using DMC-200 Controller

Since the number of instructions is large, the two controllers DMC-200 should be programmed "on the fly." This requires the usage of the jog mode of the controller. In the jog mode, the motor speed and acceleration are commanded "on the fly." Other useful commands are the following:

- IS - INCREMENT SPEED command changes the slew speed by the increment specified.

- AP - The AFTER POSITION command is a trip point used during sequences to control the timing of events. This command delays the execution of the following command until the absolute actual position of the motor crosses the position specified.

6/ CONCLUSIONS

This report presents the mathematical model of a mobile robotic cart as well as the selection of dc motors, cart control systems, and the definition of the mobile robot functions associated with a hardware and software suitable for the application of cleaning. With this work the autonomous mobile cart is completely defined and may serve as a reference for the actual implementation of its hardware and software design.

7/ RECOMMENDATIONS

The development of artificial intelligence using sensors for the autonomous operation of the mobile robotic cart for cleaning requires the completion of the following tasks:

- computer program for the trajectory planning task, based on the mathematical model previously developed;
- trajectory error correction module: develop a mathematical model and computer program;
- develop software to control and collect data from the guidance system;
- main computer requirements: speed, memory, dimensions, and peripherals;
- real-time program for the motion control of the robot;
- testing and validation of the working robotic cart.

8/ REFERENCES

[1] Galil Motion Control, DMC-200, 2-Axis Motion Controller, User's Manual, Galil Motion Control, Mountain View, CA 94043.

[2] Handbook of Tables for Applied Engineering Services.

Table 1/ Trajectory Steps

TRAJECTORY SEQUENCE	TRAJECTORY SEGMENT	CART MOTION TYPE	DIRECTION OF MOTION (nt, nr FLAGS)	LENGTH OF THE MOTION STEP (mm or rad)	WAITING TIME BETWEEN TWO ADJACENT TRAJECTORY SEQUENCES (seconds)	TASK OF THE MAIN TOOL (LOCUS OF POINT P_1)	TASK OF THE SECONDARY TOOL (LOCUS OF POINT P_2)	C_i
1	1 2 3	transl. transl. transl.	1 - forward 1 - forward 1 - forward	L_1 720.0 352.5	between sequences 1 and 2: 0	middle middle middle	retracted retracted retracted	C_1 C_2 C_3
2	1	rotation	1 - left	$\frac{\pi}{2}$	n1	P_1 moved left, l_A (mm)	P_2 extended	—
3	1	transl.	1 - forward	576.5	n2	kept at the previous position	kept at the previous position	C_2
	2	transl.	1 - forward	3.5		moved left 195.0	kept at the previous position	—
. 49								

Table 2/ Integration

TRAJECTORY SEQUENCE	TRAJECTORY SEGMENT	MOTORS												TIME OF MOTION
		1			2			3			4			
		θ	ω	ε	θ	ω	ε	θ	ω	ε	θ	ω	ε	
1	1													
	2													
	3													
2	1													
3	1													
	2													
. 49														

Table 3/ Mobile Cart Specifications

Mobile Cart	
Mass	200 kg (440 g-lb)
Moment of inertia	2.6021 kg m^2 (8891.6 lb-in^2)
Centre of mass wrt driving axle	0.178 m (7 in)
Distance between driving wheels	0.228 m (9 in)
Distance between casters and centre of mass	0.178 m (7 in)
Driving wheel radius	0.08255 m (3.25 in)
Slew speed of cart	0.5 m/s (1.64 ft/s)
Acceleration/deceleration of cart	1.0 m/s^2 (3.28 ft/s^2)
Rolling friction between wheels and car floor	0.01
Required peak torque	9.868 Nm (87.49 in-lb)
Required rated torque	1.962 Nm (17.39 in-lb)
Required speed of the shaft	63.69 rpm
Required acceleration of the shaft	13.33 rad/s^2

Note: m_p = 200 kg (440.9 lb) ensures a good provision for any equipment (in terms of mass)

Table 4/ Motor Specifications

U9FGHD DC Gearmotor	
Gear ratio	79.2:1
Rated speed	38 rpm
Peak torque	13.22 Nm (11.7 in-lb)
Rated torque	5.61 Nm (49.7 in-lb)
Rated current	5.2 A
Rated voltage	12.9 V
Power out	22 W
Weight	1.45 kg (3.2 lb)
Length (with encoder)	0.076 m (3 in)
Diameter	0.095 m (3.75 in)
Cost (with N = 500 encoder pulses/rev)	≈ $430 US

Note: 1/ PMI gearmotors have a backlash of a couple of degrees (REVEX gearmotors have only a few minutes of backlash, but they are heavier, bigger, and more expensive).

2/ The motor rated speed is 38 rpm < 63.69 rpm. The 38 rpm rated speed corresponds to 0.3 m/s (1 ft/s), which is acceptable for the cleaning task.

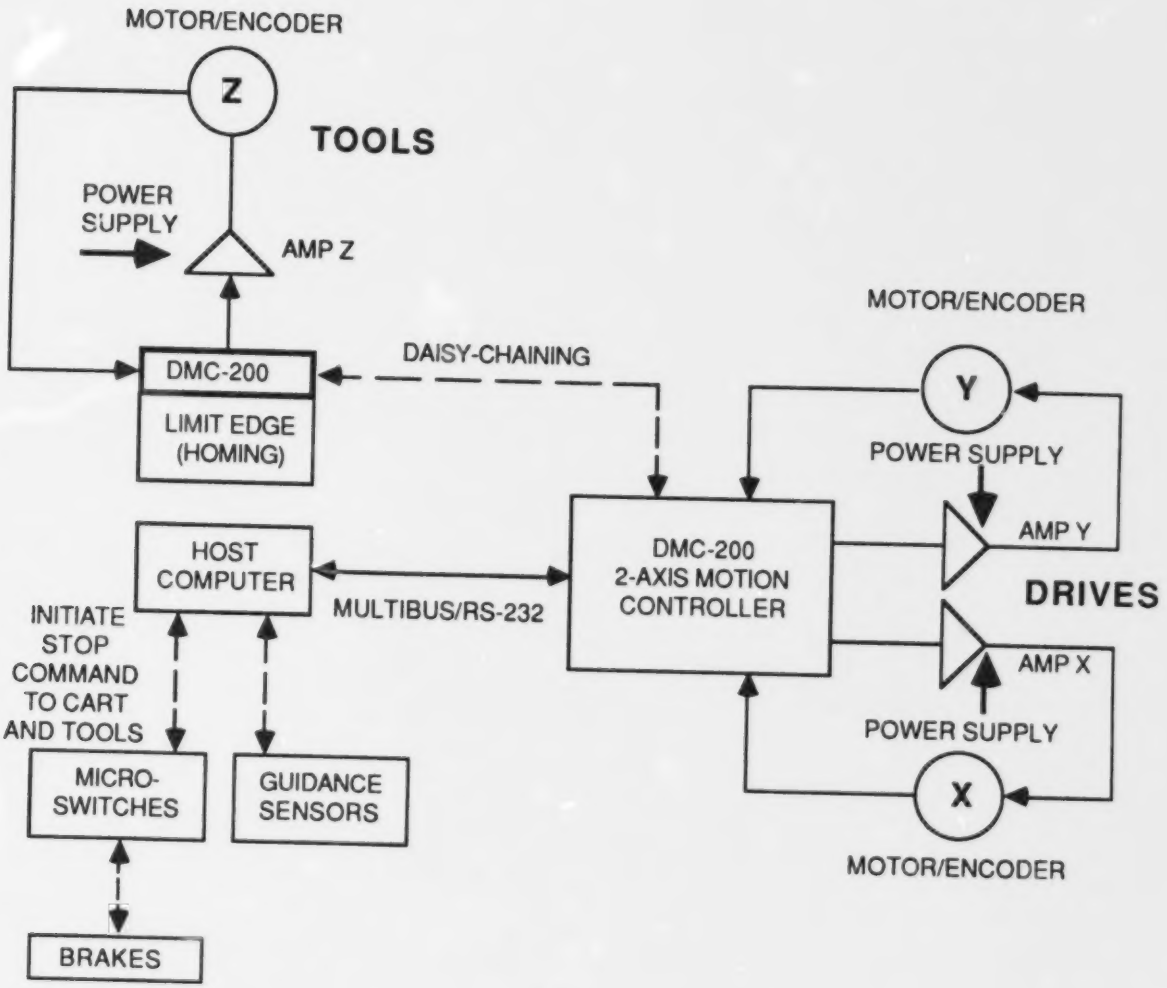


Figure 1/ Block Diagram of the Motion Control System

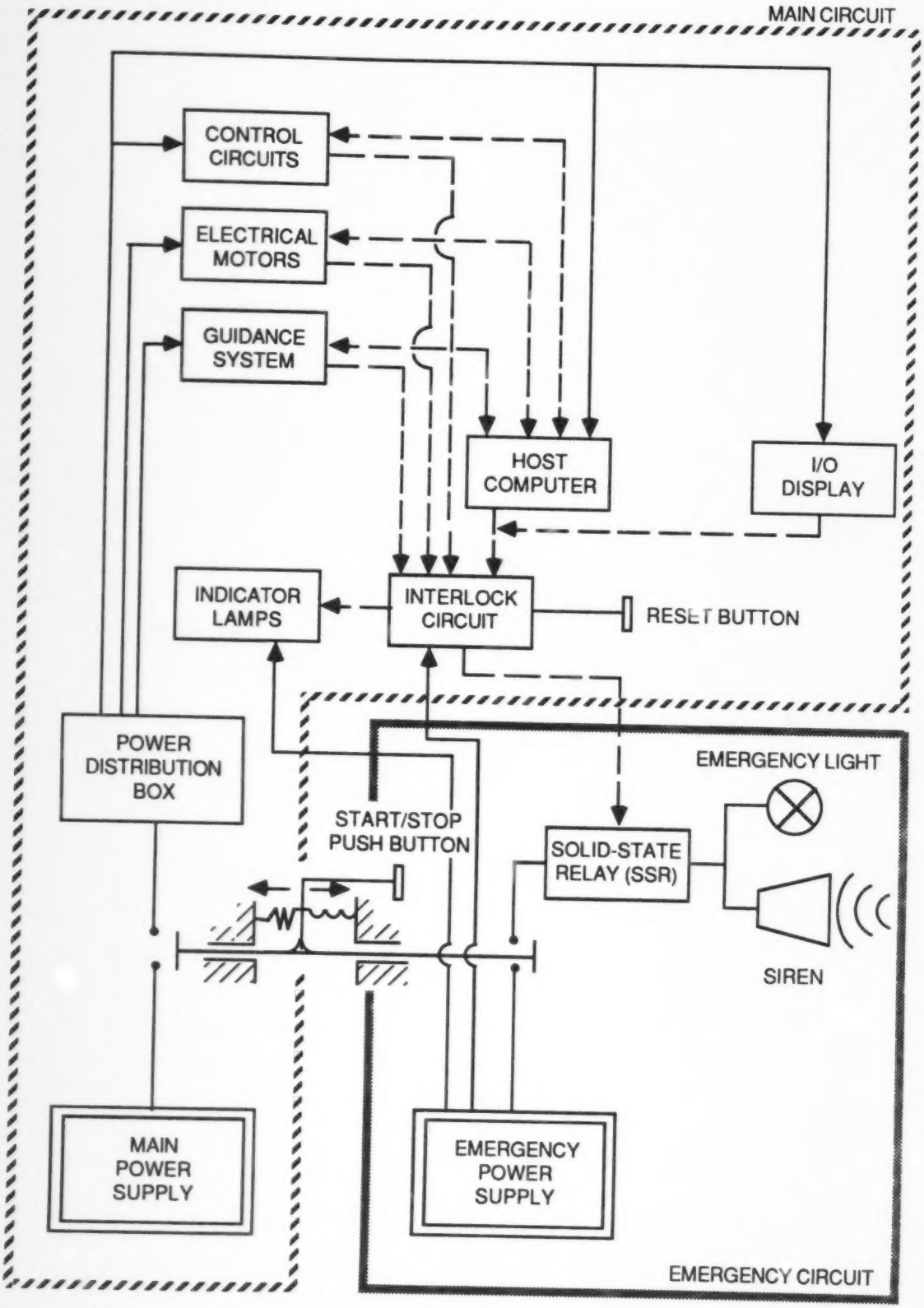


Figure 2/ Logical Layout of the Robotic Circuits

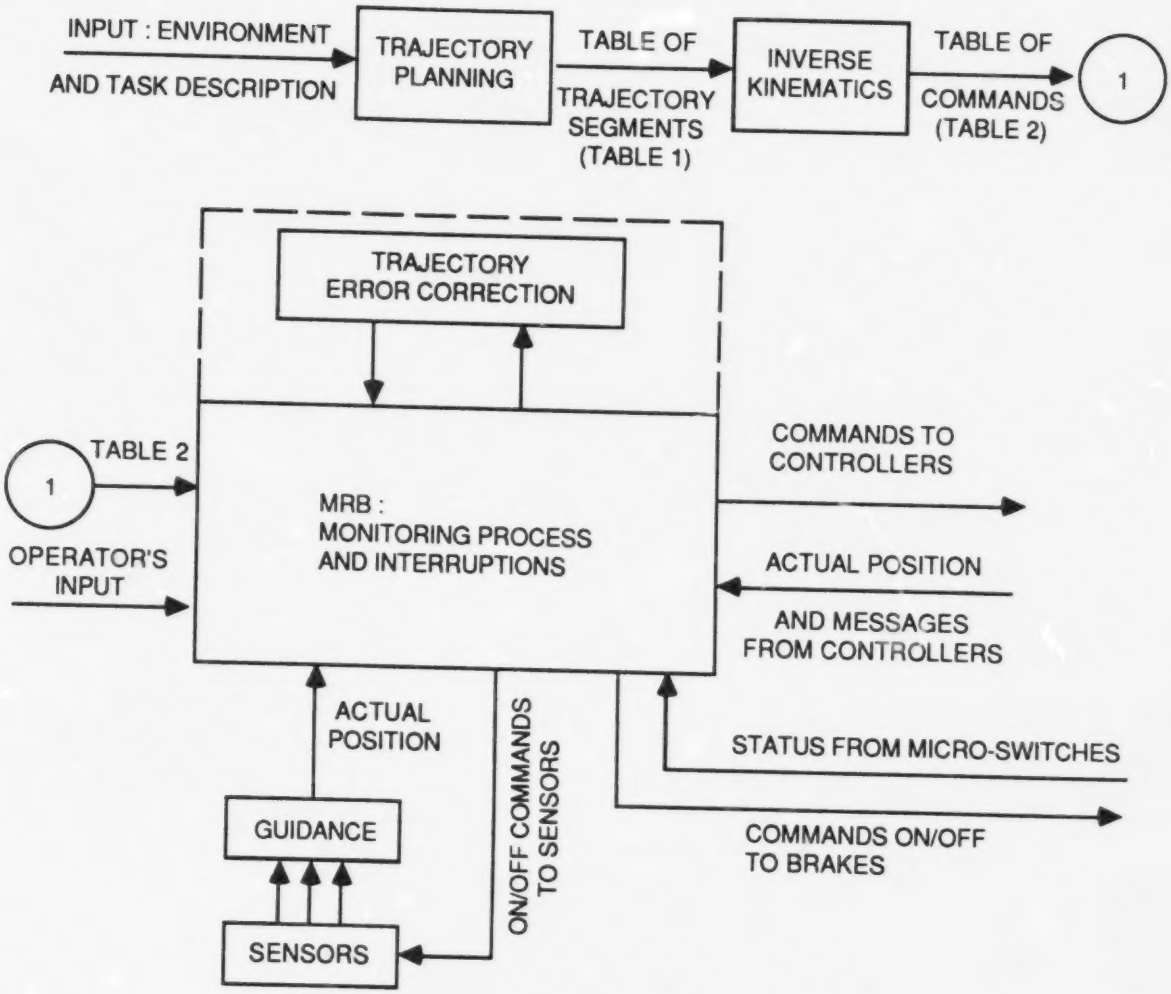


Figure 3/ Robotic Information Flow

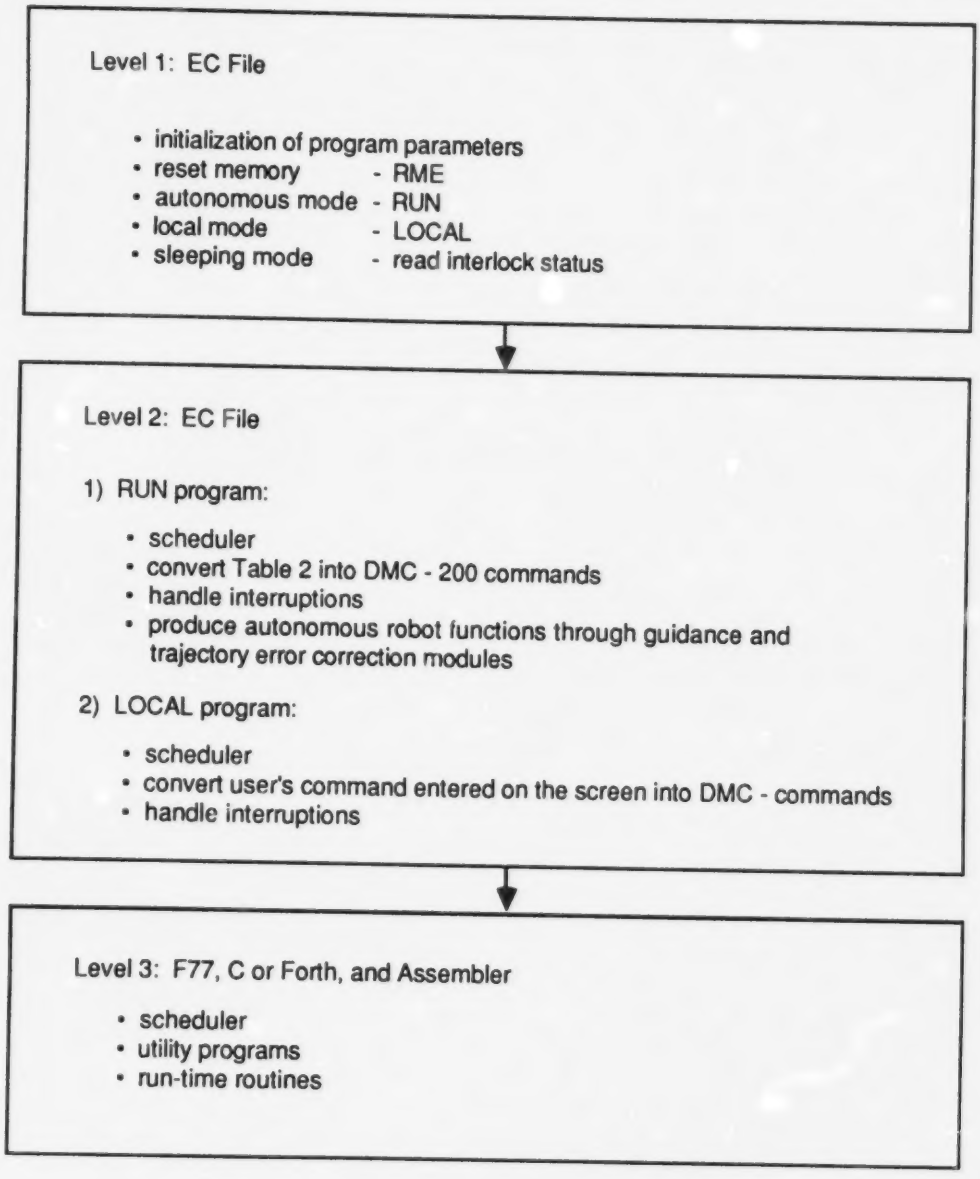

Level 1: EC File
• initialization of program parameters
• reset memory - RME
• autonomous mode - RUN
• local mode - LOCAL
• sleeping mode - read interlock status
Level 2: EC File
1) RUN program:
• scheduler
• convert Table 2 into DMC - 200 commands
• handle interruptions
• produce autonomous robot functions through guidance and trajectory error correction modules
2) LOCAL program:
• scheduler
• convert user's command entered on the screen into DMC - commands
• handle interruptions
Level 3: F77, C or Forth, and Assembler
• scheduler
• utility programs
• run-time routines

Figure 4/ Software Architecture of the Robot

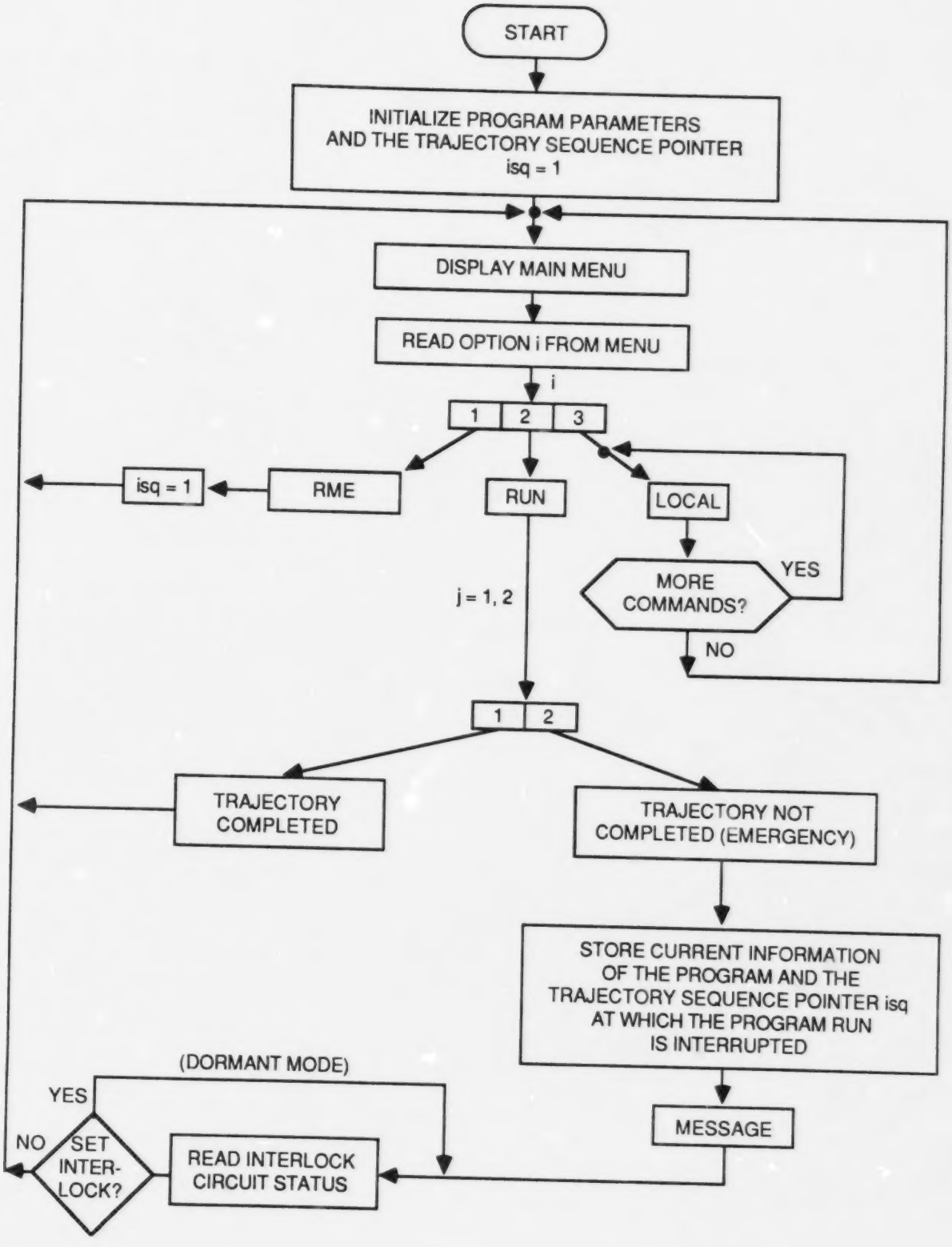


Figure 5/ First Logical Level of MRB

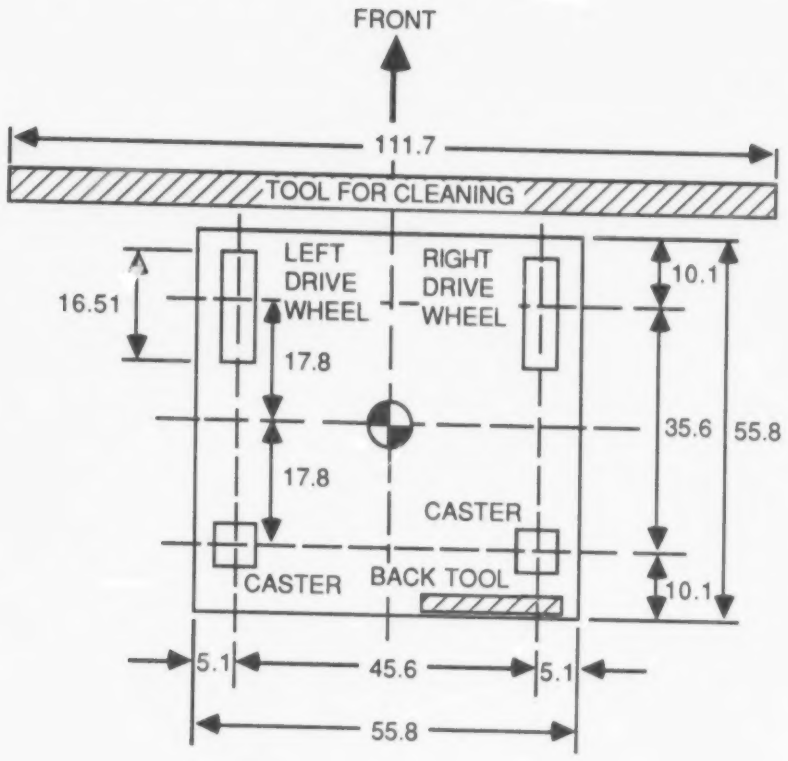


Figure 6/ Geometrical Description of the Robotic Cart (All dimensions are given in centimetres.)

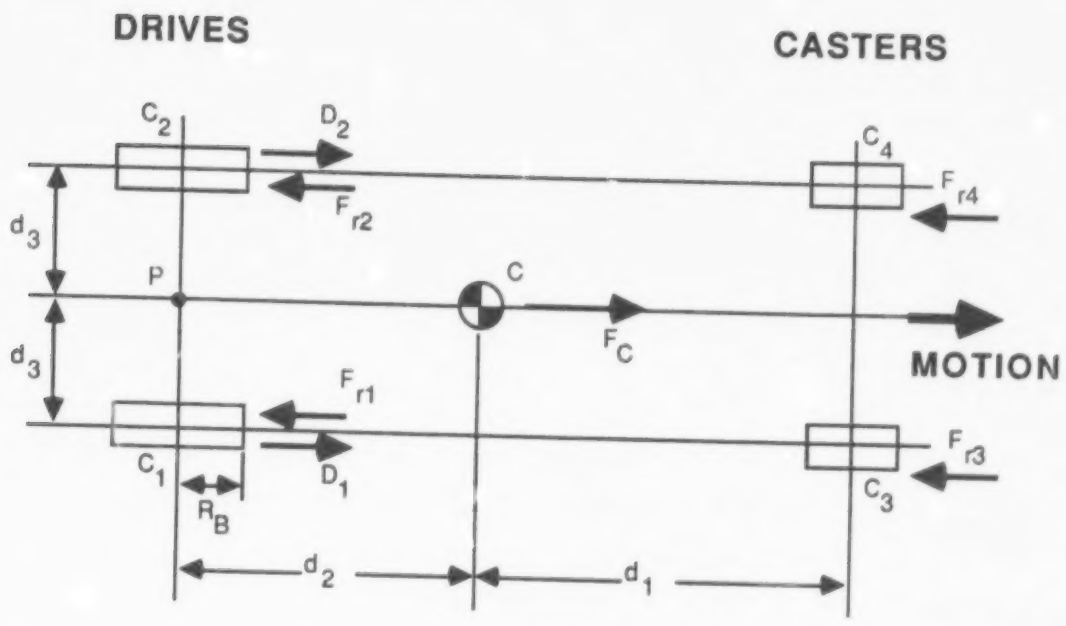


Figure 7/ Straight-Line Motion

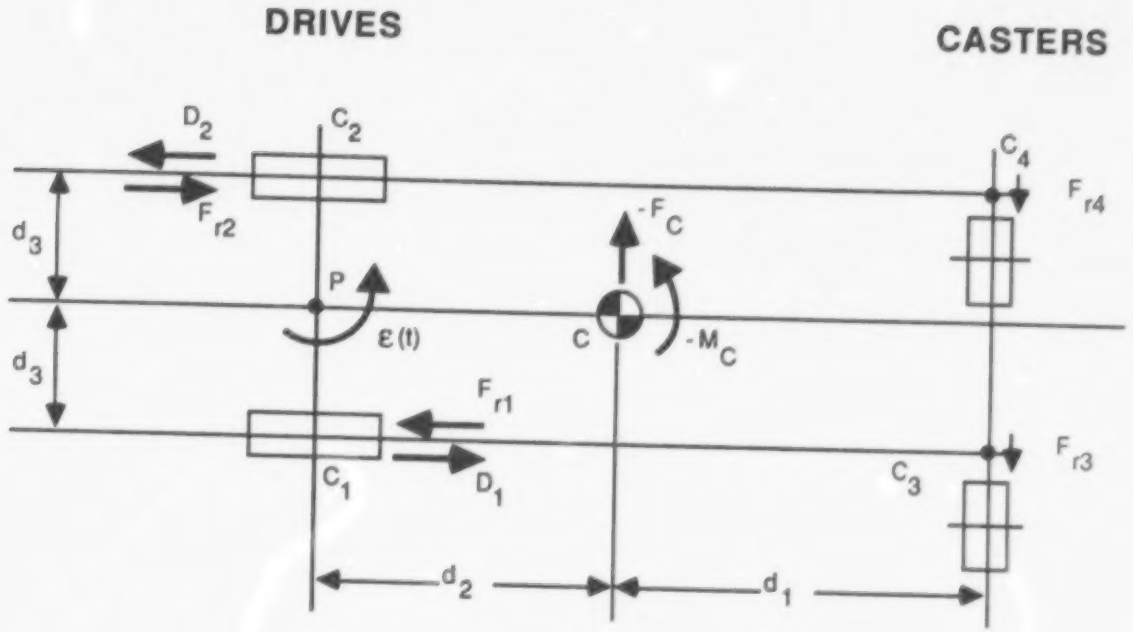
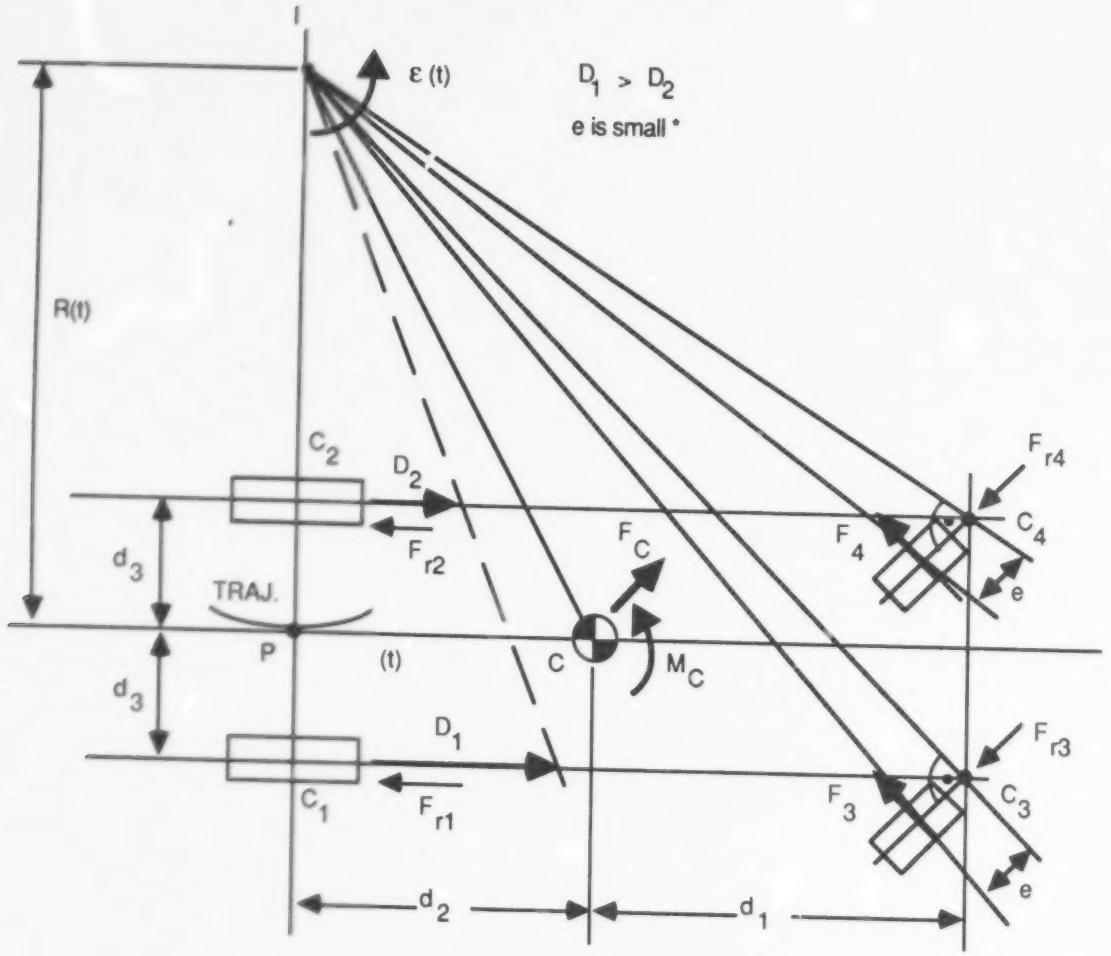


Figure 8/ Rotation in Place (About Pivoting Point P)



* $e \simeq 10\ mm << d_1 + d_2 = 558.8\ mm$

and

$F_3 e + F_4 e$ is ignored in the dynamic equations

Figure 9/ General Motion Case

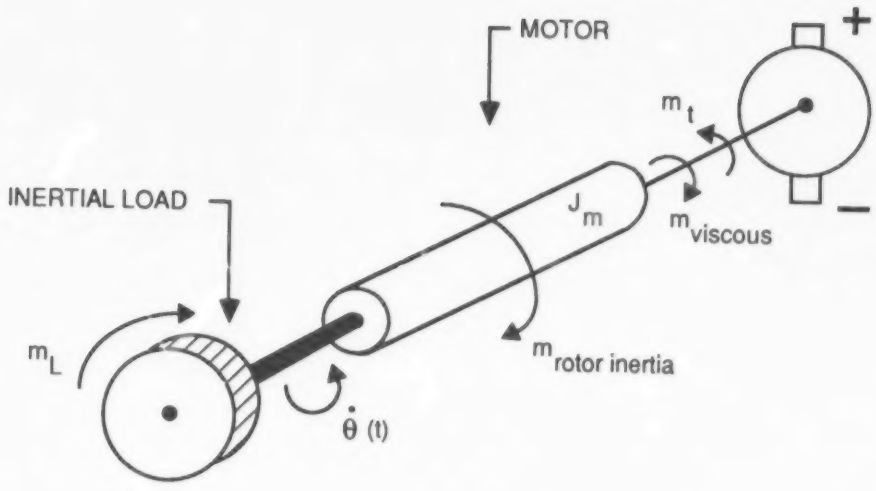
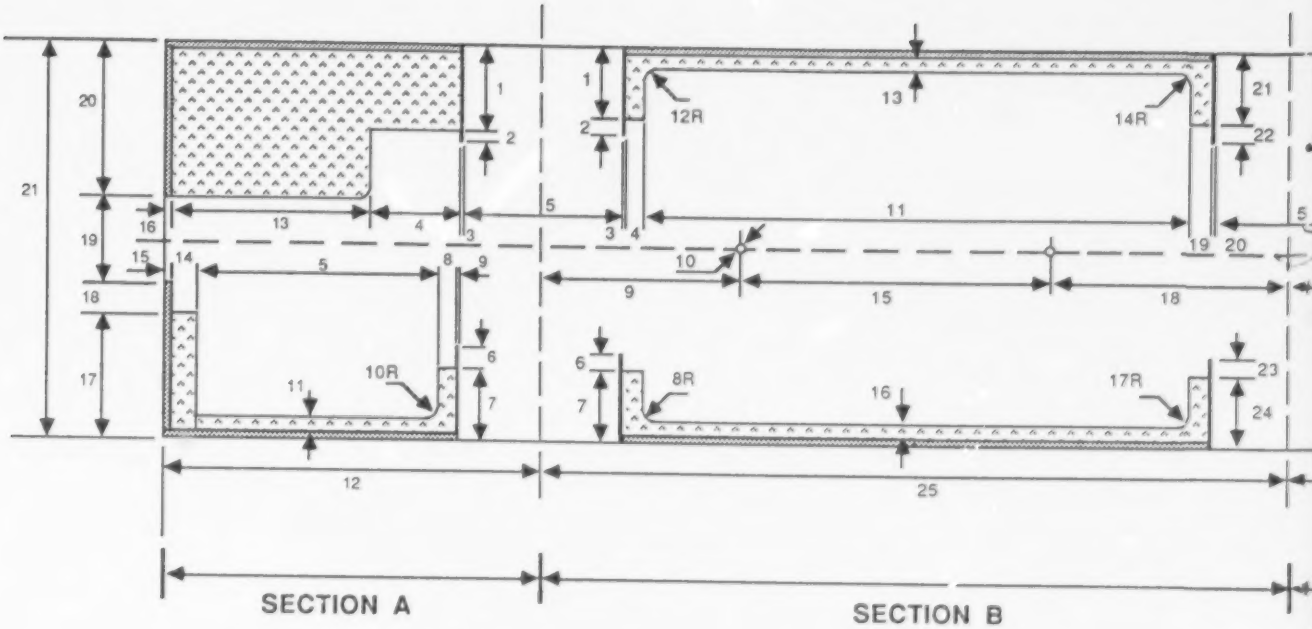


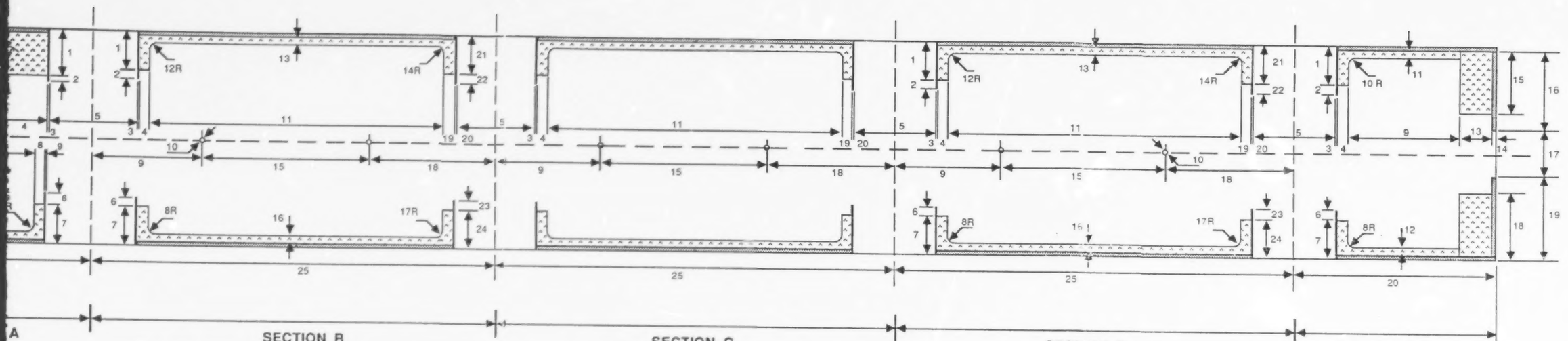
Figure 10/ DC Motor Loaded with the Inertial Load, m_L



1	590	14	215
2	90	15	60
3	9	16	60
4	661	17	895
5	1790	18	220
6	195	19	705
7	525	20	1125
8	140	21	2945
9	7	22	
10	500	23	
11	110	24	
12	2783	25	
13	1495		

1	530	14	500
2	140	15	2368
3	9	16	115
4	135	17	500
5	1146	18	1504
6	150	19	135
7	525	20	9
8	500	21	530
9	1509	22	140
10	38	23	140
11	4178	24	530
12	500	25	5381
13	115		

Figure 11/ Subway Car Schematic



SECTION A

	215
	60
	60
	895
	220
	705
	1125
	2945

SECTION B

1	530	14	500
2	140	15	2368
3	9	16	115
4	135	17	500
5	1146	18	1504
6	150	19	135
7	525	20	9
8	500	21	530
9	1509	22	140
10	38	23	140
11	4178	24	530
12	500	25	5381
13	115		

SECTION C

1		14	
2		15	2368
3	9	16	
4	135	17	
5	1141	18	1504
6		19	135
7		20	9
8		21	
9	1504	22	
10		23	
11	4178	24	
12		25	5376
13			

SECTION D

1	530	14	500
2	140	15	2368
3	9	16	115
4	135	17	500
5	1141	18	1504
6	140	19	135
7	530	20	9
8	500	21	530
9	1504	22	140
10	38	23	140
11	4178	24	530
12	500	25	5376
13	115		

SECTION E

1	530	14	60
2	140	15	865
3	9	16	1125
4	135	17	700
5	1146	18	865
6	140	19	1125
7	530	20	2780
8	500	21	
9	1535	22	
10	500	23	
11	115	24	
12	115	25	
13	470		

ar Schematic

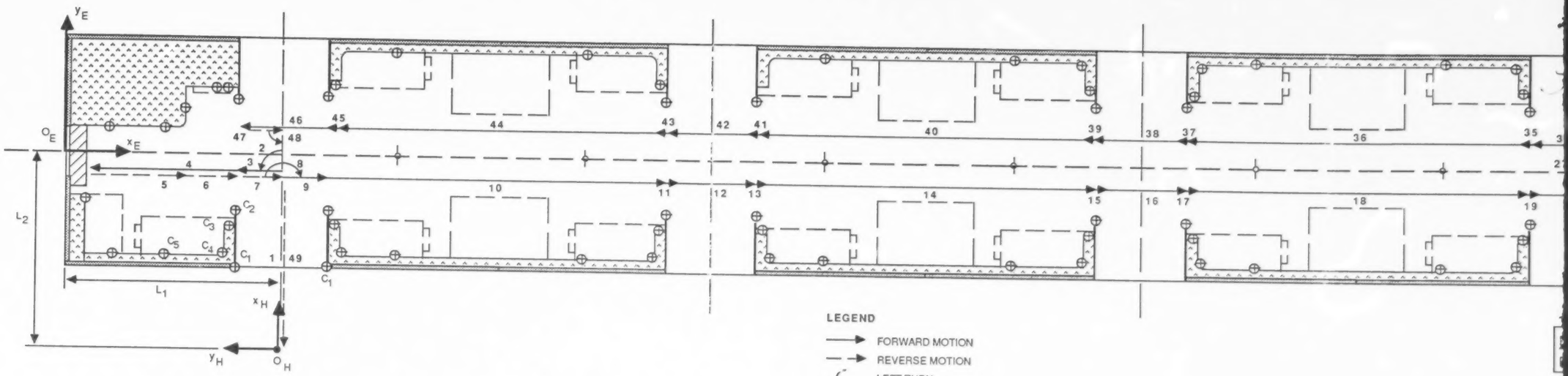
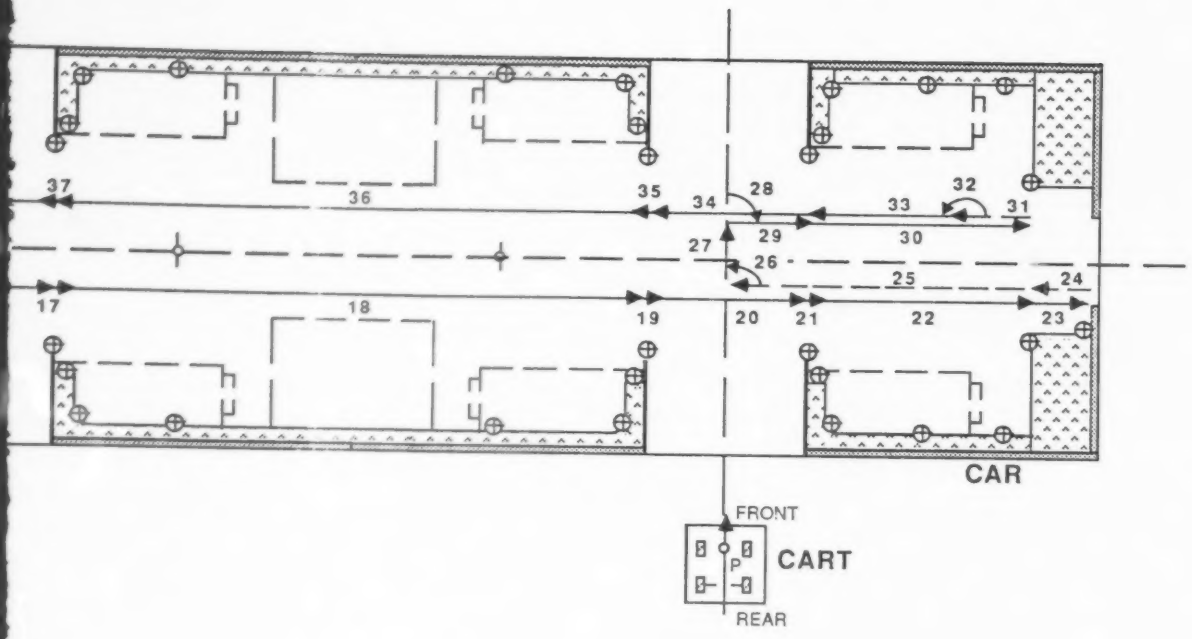


Figure 12/ Working Envelope, Cart Trajectory


37
36
35
34
28
33
32
31
29
30
27
26
25
24
17
18
19
20
21
22
23
CAR
FRONT
P
CART
REAR

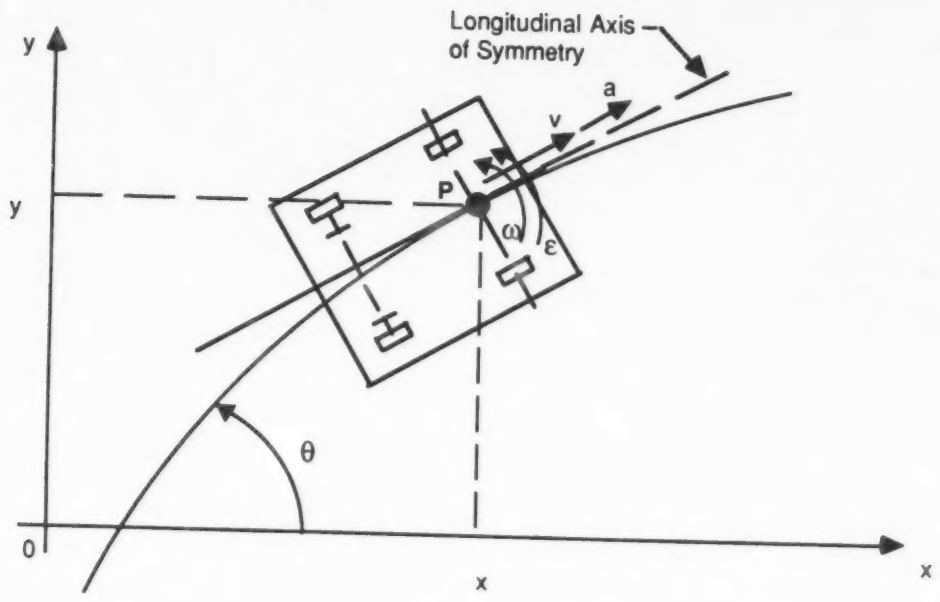


Figure 13/ Motion Characteristics of the Robotic Cart

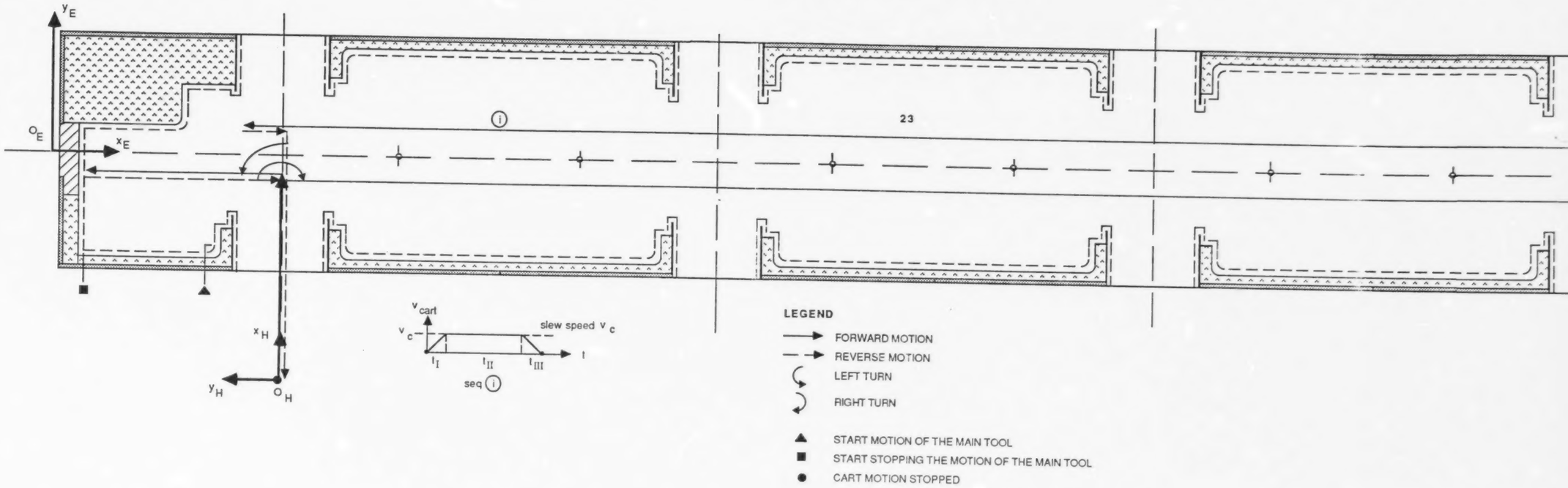
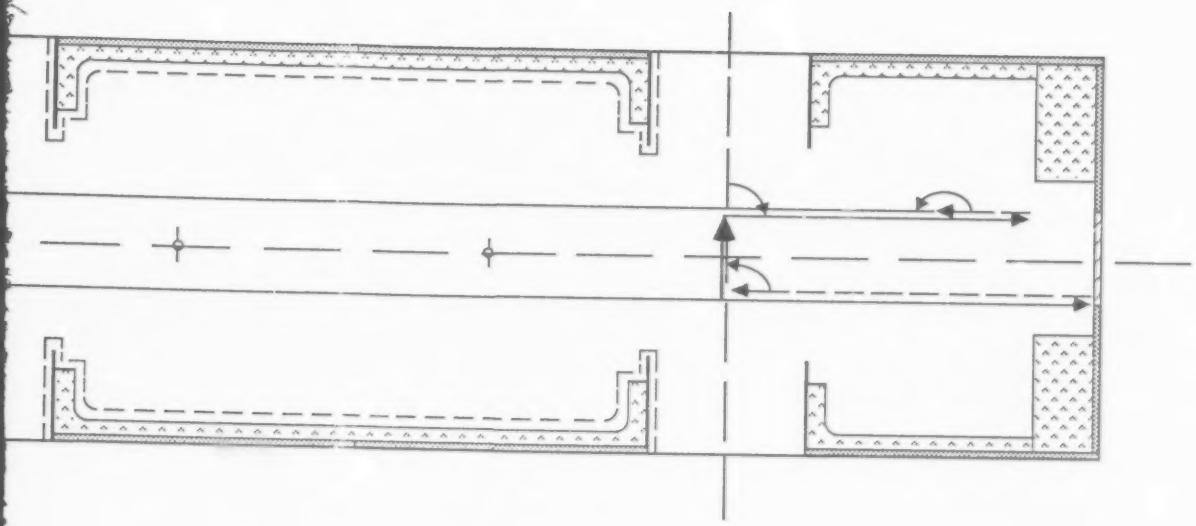


Figure 14/ Working Envelope, Floor Covering



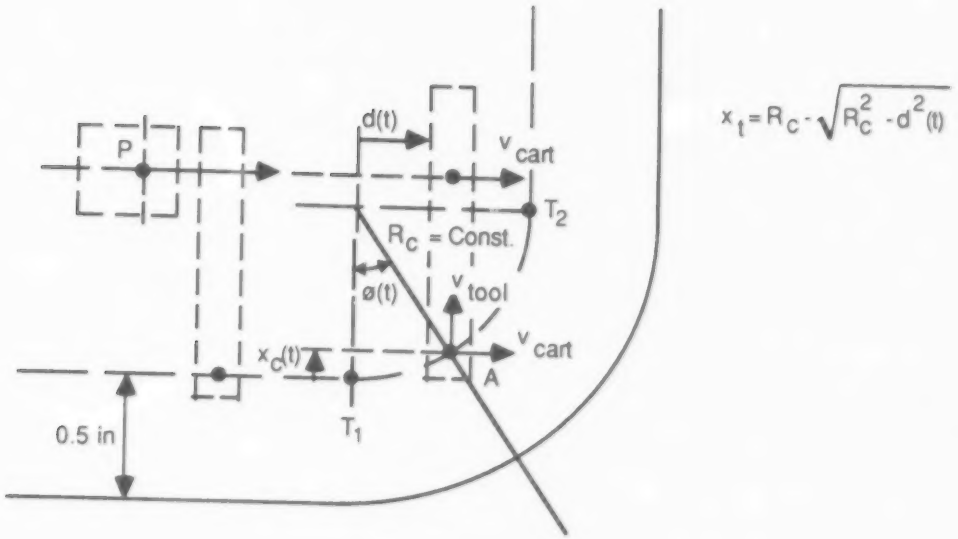


Figure 15/ Corner -- Concave Circle

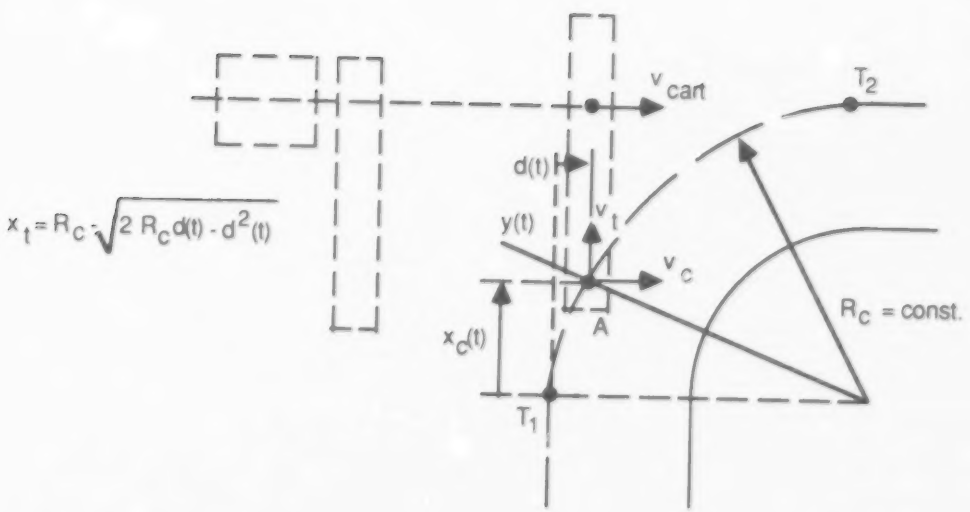


Figure 16/ Corner -- Convex Circle

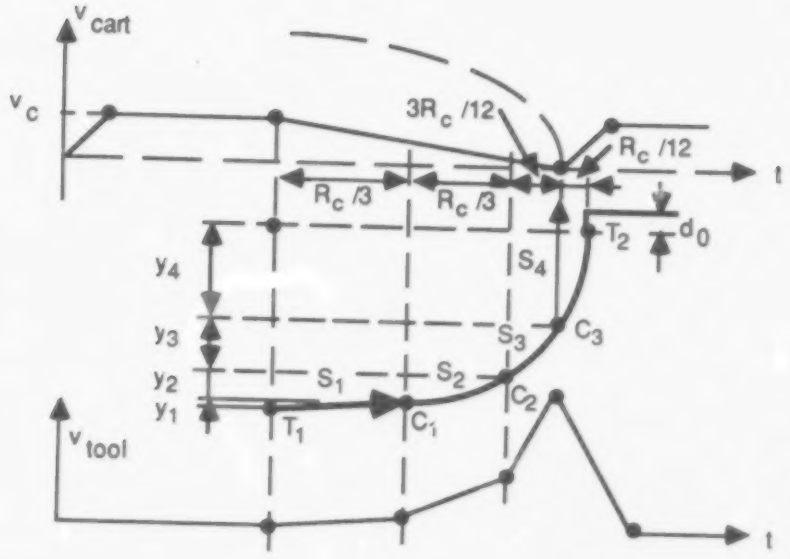


Figure 17/ Discretization of Motion for the Concave Corner, Segment by Segment

APPENDIX

A/ $\circledcirc I_{zz}$ calculation

The mechanical moment of inertia of the cart may be obtained using an appropriate formula from any mechanical handbook or may be calculated as follows:

$$\circledcirc I_{zz} = \int (x^2 + y^2)dm,$$

$$dm = \int dxdxdydz$$

If:

h = rectangular parallelepiped height
a = rectangular parallelepiped length
b = rectangular parallelepiped width

Then:

$$\circledcirc I_{zz} = \int_{-h/2}^{h/2} dz \int_{-a/2}^{a/2} dy \int_{-b/2}^{b/2} (x^2 + y^2)dx = \int bhl \frac{a^2 + b^2}{12} = m_p \frac{a^2 + b^2}{12}$$

B/ Encoder frequency ν

$$\nu = \frac{1 - \frac{\Delta S}{90}}{4T_p} \text{ [Hz]}$$

where:

ΔS = encoder state width error (electrical degrees)
T_p = program sampling period (s)